



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**DECEPTIVE TACTICS FOR PROTECTING CITIES AGAINST  
VEHICLE BORNE IMPROVISED EXPLOSIVE DEVICES**

by

Manuel Xavier Lugo

March 2008

Thesis Advisor:  
Second Reader:

Javier Salmeron  
Paul Sanchez

**Approved for public release; distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> March 2008	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> Deceptive Tactics for Protecting Cities Against Vehicle Borne Improvised Explosive Devices			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Manuel Xavier Lugo				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b> <p>This thesis focuses on interdiction of Vehicle Borne Improvised Explosive Devices (VBIED) on a major city by using "transparent" and "deceptive" assets. Transparent assets (e.g., road blocks) are those for which we assume positions are known by both attackers and interdictors. "Decoys" and "traps" are deceptive assets. Decoys are meant to be perceived as effective interdiction assets by attackers, while traps are not perceived. We use a mathematical optimization model to allocate interdiction assets maximizing expected interdicted "value." Then, we use agent-based simulation to assess the effectiveness of those interdiction plans against a variety of attacker's behaviors: perceptive (as assumed by the optimization), naïve, communicative, route blocker (static), route blocker (dynamic) and clairvoyant. We use two test networks and seven scenarios consisting of different combinations of interdiction assets. From our analysis we note that: (a) if the network incorporates deception, any behavior other than perceptive may be advantageous to the attacker; (b) a communicative behavior proves effective for the attackers against scenarios containing traps; (c) decoys are most effective if used in defense against perceptive-like behaviors; and, (d) if the defender expects perceptive-like behavior, then adding transparent assets to traps and decoys may be of little value.</p>				
<b>14. SUBJECT TERMS</b> Deception, Network Interdiction, Agent Based Simulation, Shape file to arcs			<b>15. NUMBER OF PAGES</b> 91	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited.**

**DECEPTIVE TACTICS FOR PROTECTING CITIES AGAINST VEHICLE  
BORNE IMPROVISED EXPLOSIVE DEVICES**

Manuel X. Lugo  
Lieutenant Commander, Supply Corps, United States Navy  
B.S. ME, Georgia Institute of Technology, 1996

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN OPERATIONS RESEARCH**

from the

**NAVAL POSTGRADUATE SCHOOL  
March 2008**

Author: Manuel Xavier Lugo

Approved by: Dr. Javier Salmeron  
Thesis Advisor

Dr. Paul Sanchez  
Second Reader

James N. Eagle  
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

This thesis focuses on interdiction of Vehicle Borne Improvised Explosive Devices (VBIED) on a major city by using “transparent” and “deceptive” assets. Transparent assets (e.g., road blocks) are those for which we assume positions are known by both attackers and interdictors. “Decoys” and “traps” are deceptive assets. Decoys are meant to be perceived as effective interdiction assets by attackers, while traps are not perceived. We use a mathematical optimization model to allocate interdiction assets maximizing expected interdicted “value.” Then, we use agent-based simulation to assess the effectiveness of those interdiction plans against a variety of attacker’s behaviors: *perceptive* (as assumed by the optimization), *naïve*, *communicative*, *route blocker (static)*, *route blocker (dynamic)* and *clairvoyant*. We use two test networks and seven scenarios consisting of different combinations of interdiction assets. From our analysis we note that: (a) if the network incorporates deception, any behavior other than *perceptive* may be advantageous to the attacker; (b) a *communicative* behavior proves effective for the attackers against scenarios containing traps; (c) decoys are most effective if used in defense against *perceptive*-like behaviors; and, (d) if the defender expects *perceptive*-like behavior, then adding transparent assets to traps and decoys may be of little value.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>CONCEPT DEVELOPMENT: OVERVIEW.....</b>	<b>1</b>
<b>B.</b>	<b>THE PROBLEM IN THE CONTEXT OF NETWORK INTERDICTION MODELS.....</b>	<b>2</b>
<b>C.</b>	<b>THESIS OUTLINE.....</b>	<b>5</b>
<b>II.</b>	<b>MODELING APPROACH.....</b>	<b>7</b>
<b>A.</b>	<b>PROBLEM STATEMENT AND DATA ELEMENTS.....</b>	<b>7</b>
<b>B.</b>	<b>OPTIMIZATION MODEL.....</b>	<b>11</b>
<b>C.</b>	<b>SIMULATION.....</b>	<b>16</b>
1.	Behaviors.....	16
2.	Simulation Flow Charts.....	18
<b>III.</b>	<b>COMPUTATIONAL RESULTS.....</b>	<b>27</b>
<b>A.</b>	<b>SCENARIOS DESIGN AND RELATIONSHIPS.....</b>	<b>27</b>
<b>B.</b>	<b>BASIC PROBLEM.....</b>	<b>29</b>
1.	Description.....	29
2.	Results.....	30
<b>C.</b>	<b>DYSTOPIA.....</b>	<b>39</b>
1.	Description.....	39
2.	Results.....	43
<b>IV.</b>	<b>GRAPHICAL USER INTERFACE.....</b>	<b>45</b>
<b>A.</b>	<b>DESIGN.....</b>	<b>45</b>
1.	Tables, Queries and Macros.....	45
2.	Forms.....	48
<b>B.</b>	<b>SHAPE FILE TO ARC CONVERTER.....</b>	<b>56</b>
<b>V.</b>	<b>CONCLUSIONS AND FUTURE RESEARCH.....</b>	<b>61</b>
<b>A.</b>	<b>CONCLUSIONS.....</b>	<b>61</b>
<b>B.</b>	<b>FUTURE WORK.....</b>	<b>62</b>
<b>APPENDIX: DYSTOPIA GRAPHICS.....</b>		<b>65</b>
<b>LIST OF REFERENCES.....</b>		<b>69</b>
<b>INITIAL DISTRIBUTION LIST.....</b>		<b>71</b>

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	<i>Perceptive Behavior Flow Diagram</i> .....	20
Figure 2.	<i>Naïve Behavior Flow Diagram</i> .....	21
Figure 3.	<i>Communicative Behavior Flow Diagram</i> .....	22
Figure 4.	<i>Route Blocker Static Behavior Flow Diagram</i> .....	23
Figure 5.	<i>Route Blocker Dynamic Behavior Flow Diagram</i> .....	24
Figure 6.	<i>Clairvoyant Behavior Flow Diagram</i> .....	25
Figure 7.	Basic-Problem Network Scenario I .....	29
Figure 8.	Basic Problem, Scenario II, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	32
Figure 9.	Basic Problem, Scenario III, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	33
Figure 10.	Basic Problem, Scenario IV, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	34
Figure 11.	Basic Problem, Scenario V, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	35
Figure 12.	Basic Problem, Scenario VI, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	36
Figure 13.	Basic Problem, Scenario VII, <i>Naïve</i> and <i>Perceptive Behaviors</i> .....	37
Figure 14.	Communicative Behavior Results for Different Periods per Scenario .....	38
Figure 15.	Dystopia .....	39
Figure 16.	Dystopia in Shape File Format .....	40
Figure 17.	Cape Hazard in Shape File Format .....	40
Figure 18.	Dystopia Converted into Nodes and Arcs.....	41
Figure 19.	Cape Hazard with Attackers and Targets.....	42
Figure 20.	GUI: Supporting Table List .....	46
Figure 21.	GUI: Supporting Query List .....	47
Figure 22.	GUI: Supporting Macro List.....	48
Figure 23.	GUI: Supporting Form List.....	48
Figure 24.	Switchboard Form.....	49
Figure 25.	Forward Star Form for Network Data Input .....	51
Figure 26.	Property Table Form .....	52
Figure 27.	Agent List Form.....	53
Figure 28.	Xpress MP control form.....	54
Figure 29.	Simulation Form .....	55
Figure 30.	Network Drawer: Color Key.....	56
Figure 31.	Visualization of Dystopia with Shape Viewer 1.20.....	57
Figure 32.	Dystopia (transformed) with no intersects.....	59
Figure 33.	Final Transformation of Dystopia.....	60
Figure 34.	Dystopia Scenario II, Transparent Interdiction Only.....	65
Figure 35.	Dystopia Scenario III, Transparent Interdiction and Decoys.....	66
Figure 36.	Dystopia Scenario IV & V, Transparent Interdictions and Traps.....	66
Figure 37.	Dystopia Scenario VI & VII, Traps only .....	67

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 1.	Node Elements .....	7
Table 2.	Arc Elements.....	8
Table 3.	Arc Types Table Elements.....	9
Table 4.	Values Stored in the <i>Current Methods</i> Table .....	9
Table 5.	Agent Data Elements .....	10
Table 6.	Simulation Behaviors and Characteristics .....	18
Table 7.	General Scenario Scheme .....	27
Table 8.	Basic Problem Attacker Table .....	30
Table 9.	Result Summary for Basic Problem.....	31
Table 10.	Communicative Behavior Results for three Communication Periods .....	38
Table 11.	Dystopia Scenario Agent Table .....	42
Table 12.	Result Summary for Dystopia Scenarios .....	43
Table 13.	Excerpt Output of Shape Viewer 1.20 .....	58
Table 14.	Basic Node and Arc Construct.....	58

THIS PAGE INTENTIONALLY LEFT BLANK

## **LIST OF ABBREVIATIONS**

ABS	Agent Based Simulation
ANA	Agent Network Attack
DVF	Detection Value Function
GIS	Geographic Information System
IED	Improvised Explosive Device
IWD	Interdiction with Deception
VBIED	Vehicle Borne Improvised Explosive Device

THIS PAGE INTENTIONALLY LEFT BLANK



## EXECUTIVE SUMMARY

Regardless of the era, the purposes of military deception have always been similar: Military deception is utilized as a means to change the tactical decisions of the enemy, whether offensive or defensive. This thesis focuses on deceptive interdiction of Vehicle Borne Improvised Explosive Devices (VBIEDs). In particular we look at how to distribute a pre-specified number of “transparent” and “deceptive” interdiction assets in a city in order to maximize the interdiction of VBIEDs.

Transparent assets (e.g., road blocks) are those for which we assume positions are known by both attackers and interdictors. Deceptive assets are further divided into “decoys” and “traps.” Decoys have little or no interdiction effectiveness, and their positions are known by both sides. However, their value lies in that their effectiveness is concealed, and actually exaggerated in order to make them seem much more effective than they are. Traps, on the other hand, are effective interdiction tools whose positions are unknown to the attackers.

We build on existing mathematical optimization models to represent the problem of allocating interdiction assets maximizing expected interdicted “value,” and develop new agent-based simulation models to assess the effectiveness of those interdiction plans against a variety of attacker’s behaviors.

Seven attacker behaviors have been explored: *perceptive*, *naïve*, *communicative*, *route blocker (static)*, *route blocker (dynamic)* and *clairvoyant*. The *perceptive* behavior (assumed by the defender’s optimization model) presumes the attacker is oblivious to deceptive assets deployed by the defender. The *naïve* behavior is solely based on shortest distance to the target. All other behaviors are *perceptive*-like, but incorporate features such as learning (*communicative*), and random detours (*route blocker static* and *dynamic*). The *clairvoyant* behavior assumes full knowledge by the attacker and is used solely for bounding purposes.

Two test networks and seven scenarios per case (each scenario consisting of different combinations of interdiction assets) have been studied. Several conclusions of

general application to any case have been derived, while others are case specific. From our analysis we note the following observations:

- If the network incorporates deception, any other behavior (than *perceptive*) may be advantageous to the attacker.
- A *communicative* behavior proves particularly effective over time for the attackers against scenarios containing traps.
- Decoys are most effective if used in defense against *perceptive*-like behaviors. They are rendered ineffective against attackers which behave *naïvely*.
- If the defender expects the attacker to act with any *perceptive*-like behavior, then the addition of transparent assets to traps and decoys may be of little value.

In addition, this work has developed a graphical user interface which integrates the optimization and simulation models. This interface aids with problem data preparation and validation, transfers appropriate data and results between models, and helps the analyst visualize the solution graphically.

## **ACKNOWLEDGMENTS**

The author would like to acknowledge the following individuals for their efforts and support in making this thesis possible.

My wife Ana Maria has been a constant source of inspiration, support, love and understanding. Neither of us fully understood what these two years were going to entail, but despite the constant time dedicated to my studies and research, Ana Maria has always been there with a smile. My daughter Catalina has reached several milestones of her life during our time here, and has given me the motivation necessary to pursue my goals. My son Manuel Felipe was born here, and with him I have also regain interest in the simple things of life that get so easily ignored by adulthood.

Professor Salmeron has been a driving force, a great mentor, and a fountain of knowledge. His guidance is what has made this research mature into a product that we are both proud of.

Professor Sanchez has always been my technical conscience. I will always remember his indulgence to my conversations, many times off topic, about computers, algorithms, java and the likes and his ability to keep me on track with reality.

The OR faculty has been a constant example of professionalism.

Mr. John Locke and Ted Lewis have shown unselfishness and willingness to help, even without us ever meeting in person. Thank you for Dystopia!

A journey like this is never traveled alone. Several classmates have been motivators, source of inspiration, and torturers, but above all friends. My special thanks go to my friends Scott Hattaway, John Baggett, Ben Abbott, Alexandros Matsopoulos, and Steve Ostoin.

THIS PAGE INTENTIONALLY LEFT BLANK

## **I. INTRODUCTION**

Deception tactics have historical references ranging from the mythical Trojan horse to Operation Fortitude during World War II, and tactics applied more recently during Operation Desert Storm [Spiller, 1992]. Regardless of the era, the purposes of military deception have always been similar: Military deception is utilized as a means to change the tactical decisions of the enemy whether offensive or defensive. The enemy could offensively plan to attack a false target (or target of little value) due to deception by the friendly force. As a defensive example, the enemy might switch the majority of their forces to the wrong border due to believing an attack will come from a different location. Sometimes deception can be used to simply give a perception of ignorance, so that the original plans are executed while the enemy is oblivious of the fact that their target is aware of their intentions and has adequately defended itself. Regardless of the intentions of the deception and the resulting effects, military deception is commonly characterized as an art.

### **A. CONCEPT DEVELOPMENT: OVERVIEW**

One of the major responsibilities of a government is to safeguard the safety of its citizens. Threatening that safety is one of the terrorists' ideal tactics to undermine governments and disrupt society at large. Major cities tend to be the battlefield where these opposing objectives meet, offering the highest level of impact for terrorists and great difficulty for governments to protect, at least without disrupting the regular lives of their citizens.

There are many methods that terrorists have used in history, and there are probably more to be seen. Currently, a common method used by terrorists and insurgents is Improvised Explosive Devices (IEDs) [Dudonis, 2005]. Whether IEDs are emplaced, human carried (i.e., suicide bombers), or vehicle borne, they are all effective at challenging civilian safety. Among these three categories, Vehicle Borne Improvised Explosive Devices (VBIEDs) typically claim most lives per occurrence due to the payload capacity of vehicles, ranging from small cars to large trucks.

This thesis focuses on interdiction of VBIEDs in a major city. In particular we look at how to distribute a pre-specified number of “transparent” and “deceptive” interdiction assets in the city in order to maximize a Detection Value Function (DVF). The DVF comprises the probability of interrupting a VBIED before it reaches its target and the value of the target itself, across a number of VBIED origins and destinations. (Remark: In what follows we shall refer to VBIEDs and the terrorists who carry them interchangeably as VBIEDs, terrorists or attackers.)

Major cities are target-rich environments for VBIED attacks. Their large populations and meshed structure of highways and roads create ideal distribution networks for such attacks. On cities with higher probability of VBIED incidents, proactive measures are normally taken in order to intercept them. These measures are usually in the form of roadblocks, in which peace officers or military personnel inspect some or all of the vehicles passing through the roadblock. The personnel conducting the inspection use a spectrum of tools [Dudonis, 2005] including visual, and canine and sensor (x-ray, infrared, or electromagnetic) inspections, among others.

We use the concept of deception in determining where to place two categories of assets: “transparent” and “deceptive,” of which the latter can be further divided into “decoys” and “traps.” Transparent assets (e.g., road blocks) are those for which we assume positions are known by both attackers and interdictors, and their effectiveness can be assumed as being equally known. Decoys have little or no interdiction effectiveness, and their positions are known by both sides. However, their value lies in that their effectiveness is concealed, and actually exaggerated so as to make them seem much more effective than they are. Traps, on the other hand, are effective interdiction tools whose positions are unknown to the attackers.

## **B. THE PROBLEM IN THE CONTEXT OF NETWORK INTERDICTION MODELS**

A city road grid can be represented mathematically as network of nodes and arcs. This representation allows for the use of “standard” network analyses such as single- and

multi-commodity network flow models (see, e.g., Ahuja et al. [1993]). These standard network problems aim to maximize network efficiency, minimize costs, and the like.

Network interdiction models (see, e.g., Wood [2003]) augment standard network theory by establishing bi-level and tri-level Stackelberg games with two players: an attacker and a defender (where the latter, in fact, usually plays a double role as defender and network operator). Assuming transparency in the information shared by these players, the tri-level “defender-attacker-defender” model, for example, posits a defender using limited (defensive) resources with which to protect select components in his system (in this context, represented as a network). His goal is to minimize the potential damage a worst-case attack on the system might cause. Then, an attacker with limited (offensive) resources solves a problem to determine which components should be disabled in order to inflict maximum damage. Finally, the defender (now acting as the network operator) operates the residual (non-interdicted) network to minimize disruption.

Brown et al., [2005, 2006] show a wide variety of applications of network interdiction theory, especially in the areas of interdiction and defense of critical infrastructure. In the bi-level “attacker-defender” (or “defender-attacker”) and tri-level “defender-attacker-defender” models, transparency translates into a unique objective function, shared by both players, with diametrically opposed goals (minimize versus maximize, or vice versa). The most important advantage of employing the transparency approach is that no player can improve his strategy without compromising the outcome: (a) if the defender chooses an alternative defensive plan, an intelligent attacker could obtain a larger disruption, provided he acts optimally; and, (b) there is no other course of action for the adversary (than the one dictated by his damage-maximization problem) which can yield a larger disruption in the system.

There are certain interdiction problems, however, in which the defender could take advantage of non-transparent interdiction assets. This thesis tests an “interdiction with deception” (IWD) model, currently under development by faculty at the Naval Postgraduate School, which may ultimately help in the allocation of limited transparent and deceptive assets to interdict VBIEDs.

The IWD model seeks an optimal placement of available interdiction assets in order to maximize the probability of detection of an attacker infiltrating a network with one or multiple potential targets. The model is extended to multiple attackers, with different values for each attacker, leading to a DVF which approximates expected interdicted value.

The IWD model relies on two key assumptions: (a) the defender knows exactly how the attacker will *perceive* transparent and deceptive interdictions on arcs, as well as arcs which have not been interdicted; and, (b) the attackers will use these perceptions (specifically, the perceived probabilities of being interdicted on each network arc) to determine optimal routes to their targets.

The use of non-transparent assets provides a field commander with a force multiplier to augment the efficiency of interdiction based on transparent assets only (such as that of road blocks). However, as opposed to the fully transparent case, the IWD model cannot guarantee that, if the attacker does not behave as expected, he will not find a better route than anticipated by the model. This begs the question: “What would happen if the plan is made according to a ‘perceptive’ behavior but the attacker behaves differently?” For this reason this thesis devises and implements an agent-based simulation (ABS) model to test the IWD solution against other possible attacker’s behaviors, namely “naive,” “road-blocker (static and dynamic),” “communicative” and “clairvoyant,” which will be described in detail later in this document.

This research evaluates the IWD and ABS models on two test cases: a small network, as a proof of concept from which we may gain insights into the effects of adding deceptive interdiction assets; and, a large network, derived from a notional city, Dystopia [Locke, 2008] to evaluate the complexity of realistically-sized problems. We test the outcomes for all of the behaviors using several defender-resource scenarios per case.

We caution the reader that it is not the goal of this research to predict VBIED attacks, but to explore the potential effectiveness of interdiction tactics which include deception under several insurgent behavior assumptions. This requires merging new,



non-transparent network interdiction models with agent-based simulation, two distinct methods of analysis that seldom are combined. In essence, we are facing a stochastic problem (given the unknown behavior of the attacker, among others). As an example of previous work which combines optimization and simulation, Sanchez and Wood [2006] build an algorithm to solve two-stage stochastic problems by optimization and simulation. They first generate multiple candidate solutions for the first-stage problem (in our case, these decisions would correspond to the placement of interdiction assets). Then, multiple second-stage outcomes are simulated (terrorist behaviors), which are formally tested to ensure the candidate solutions contain an optimal one with certain probability. While this work has not pursued Sanchez and Wood's algorithmic approach, we recognize the underlying motivation is similar, and that their methodology would be useful to analyze the problem under consideration.

Although city protection against VBIEDs is the focus area of this research, the application of the theory, methods and software developed in this thesis is much broader. For example, similar deception techniques can be adapted for other than road networks, such as communication and data, with applications to intruder detection. Also, civilian authorities involved in Homeland Defense can take advantage of similar analysis in problems such as vehicle high speed deterrence and trapping, drug interdiction and border patrol.

### **C. THESIS OUTLINE**

The remainder of the thesis is organized as follows: Chapter II presents the modeling approach, that is, the problem statement and the optimization and simulation models. Chapter III introduces the two test scenarios and associated computational experience. Chapter IV illustrates the supporting tools (database and graphical user interface) that have been developed to facilitate the analysis. Finally, Chapter V discusses our conclusions and recommends future work.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. MODELING APPROACH

This chapter discusses the modeling approach for the problem of allocating limited interdiction assets (transparent and deceptive) in a city in order to maximize a DVF which captures the probability of interdicting attacks against pre-specified targets and the value of those targets. Before we discuss the mathematical approach (optimization and simulation models), we state the problem in terms of its components and data elements.

### A. PROBLEM STATEMENT AND DATA ELEMENTS

The goal of the problem is to obtain a plan which details where in a city to place a limited number of interdiction assets whose combined effectiveness to protect pre-determined locations inside the city is maximized.

The city's roads are modeled as a network; therefore we define the network data elements in terms of nodes and arcs. The data elements related to the nodes are listed in Table 1.

Attribute	Data Type	Description
Node	Text	Name for the Node
Location X	Real Number	X-location for the Node
Location Y	Real Number	Y-location for the Node

Table 1. Node Elements

A node is uniquely identified by its name and has X-Y Cartesian location attributes. These are necessary for arc-length calculations.

Arcs are defined by their “head” and “tail” nodes, representing the starting and end nodes of the arc, respectively. Table 2 shows all the arc attributes.

Attribute	Data Type	Description
Tail	Text	Starting node
Head	Text	Ending node
Type of Arc	Text	Links to Arc type attributes
Interdictable	Yes / No	Is the Arc interdictable?
Nominal Interdiction Probability	Proportion	Calculated based on arc length and speed
Transparent Interdiction Probability	Proportion	Transparent (i.e., road blocks)
Trap Interdiction Probability	Proportion	Traps (Invisible to attackers)
Decoy Perceived Probability	Proportion	Decoy (Perceived by attackers, higher than actual)

Table 2. Arc Elements

Besides the tail and head elements, the arc data structure contains fields for the *nominal*, *transparent* and *trap* interdiction probabilities, and the *decoy* perceived probability. All of these are pre-specified by the user or pre-calculated according to certain assumptions, as explained below. Some arcs can be deemed non-interdictable, thus the need for the *Interdictable* Boolean attribute.

Each type of arc has its own attributes (see Table 3). The *Nominal Interdiction Probability* is that of interrupting the VBIED while traversing an arc even if the arc is not interdicted. This may occur, for example, if the vehicle has a breakdown and/or is recognized by routine police patrols. This value is typically very low, but it is a necessary element in our approach. The nominal interdiction probability is determined by the *NominalMultiplier* (measured in  $s^{-1}$  to reflect the fact that, the longer the vehicle stays on the road, the more likely it is to be detected), the *AvgSpeed*, which is the average vehicle speed on the road, and the *ArcLength*, which is calculated using the X-Y coordinates of the *Tail* and *Head* nodes. The *Nominal Interdiction Probability* is then calculated according to the following formula:

$$Nominal\ Interdiction\ Probability = ArcLength \frac{NominalMultiplier}{AvgSpeed}$$

For example, if a 100-meter road is given a *NominalMultiplier* value of  $0.0006 \text{ s}^{-1}$  and an *AvgSpeed* value of 20 m/s (72 km/h) then the *Nominal Interdiction Probability* assigned to this arc is 0.3%.

Attribute	Data Type	Description
Type of Arc	Text	For example (City, Highway, Rural)
Avg Speed	Real Number	Average Speed of traversal
Nominal Multiplier	Real Number	Multiplier for Nominal Interdiction Probability Calculation

Table 3. Arc Types Table Elements

Currently, the model explores three interdiction methods, different from nominal (Table 4), with a specified effectiveness for each one.

Methods				
Method Name	Method Type	Baseline Probability of Interdiction	Distance Correction Factor	Speed Correction Factor
GenericDecoy	Decoy	0.25	5	0
GenericTrap	Trap	0.75	0	0
RoadBlock	Transparent	0.25	0	0

Table 4. Values Stored in the *Current Methods* Table

Moreover, by modifying the *Method Name* and *Method Type* fields we can add more methods of each type with different effectiveness values. Table 4 shows the values assigned to the different methods explored in this research to calculate the (other than nominal) interdiction probabilities. The distance and speed correction factors are combined with the nominal probability of detection and the average speed associated with the arc in order to enhance or degrade the interdiction probability of the method. Speed, for example, could degrade the interdiction probability of a trap, if such a trap requires a certain time engaged and has a limited range of effectiveness. Similarly, arc length could potentially improve the effectiveness of a decoy, if such a decoy can be

perceived throughout the arc. Conversely, to reflect degradation, we can use negative numbers. The interdiction probability on the arc is determined as follows:

$$\begin{aligned} \text{Interdiction Probability} = \\ \max \{0, \min \{ \text{Baseline Probability of Interdiction} + \\ \text{Nominal Probability Detection} \times \text{Distance Correction Factor} - \\ \text{Average Speed} \times \text{Speed Correction Factor}, 1 \} \} \end{aligned}$$

Next, we define the attacker’s data elements (Table 5). Each attacker is given a name, value, and target and start nodes, all of which are user-defined inputs. The model assumes that the starting positions for the attackers are known but, if this is not the case, we may accommodate an unknown origin by adding artificial, non-interdictable arcs from a fictitious starting node to all the potential starting nodes for the attacker.

The attacker’s value is assumed to be the same for both the attacker and the defender. For example, it may be based on the amount of explosives carried by that attacker [BATF, 2005], therefore reflecting, to some extent, the number of people that would be killed by that VBIED. Our computational experiment assumes each attacker has a unique target; however, the optimization model presented in the next section allows for a multiple-target attacker who, based on the network configuration after interdiction, decides on his best target. Similarly, it is possible that multiple attackers start at the same origin with different destinations.

Attribute	Data Type	Description
Agent	Text	Agent's identifier
Value	Number	Target value
Target Node	Text	Target Node name
Start Node	Text	Start Node name

Table 5. Agent Data Elements

Finally, the problem data is completed by specifying the number of interdiction assets of each type available to the defender. These parameters (number of transparent, traps and decoys) are globally referred to as “interdiction resources.”

## B. OPTIMIZATION MODEL

In this section we describe the IWD model for the problem stated in Section A. The full derivation and details of the model can be found in [Salmeron, 2007]. This section summarizes the mathematical formulation and its parameters.

The notation for the optimization model is as follows:

### Sets and indices

$I$ , set of nodes in the network, for  $i, j \in I$

$A \subset I \times I$ , set of arcs in the network, for  $(i, j) \in A$

$N$ , set of potential attackers trying to cross through the network, for  $n \in N$

$s_n \in I$ , source node for attacker  $n$ .

$T_n \subset I$ , subset of possible target nodes for attacker  $n$ . We assume  $s_n \notin T_n$

### Parameters

$v_n$ , value of attacker  $n$  (e.g., based on the amount of explosive he carries)

$p_{nij}$ , nominal probability of interdicting attacker  $n$  traversing (non-interdicted) arc  $(i, j)$ . Note:  $q_{nij} = 1 - p_{nij}$

$\tilde{p}_{nij}$ , probability of interdicting attacker  $n$  traversing arc  $(i, j)$  with transparent interdiction asset. Note:  $\tilde{q}_{nij} = 1 - \tilde{p}_{nij}$

$\bar{p}_{nij}$ , non-transparent probability of interdicting attacker  $n$  traversing arc  $(i, j)$  with trap interdiction asset. (Attackers perceive nominal.) Note:  $\bar{q}_{nij} = 1 - \bar{p}_{nij}$

$\overline{\bar{p}}_{nij}$ , non-transparent probability of interdiction perceived by attacker  $n$  while traversing “decoy” arc  $(i, j)$ . (Actual is nominal.) Note:  $\overline{\bar{q}}_{nij} = 1 - \overline{\bar{p}}_{nij}$

$\tilde{R}$ , amount of transparent interdiction resource

$\bar{R}$ , amount of trap interdiction resource

$\overline{\bar{R}}$ , amount of decoy interdiction resource

$\tilde{a}_{ij}$ , amount of transparent interdiction resource needed to interdict arc  $(i, j)$

$\bar{a}_{ij}$ , amount of trap interdiction resource needed to interdict arc  $(i, j)$

$\overline{\bar{a}}_{ij}$ , amount of decoy interdiction resource needed to interdict arc  $(i, j)$

### Decision variables

$x_{nij}$ , 1 if attacker  $n$  uses arc  $(i, j)$ ; 0 otherwise. The vector of all  $x_{nij}$  is denoted as  $\mathbf{x}$ , and the vector of all  $x_{nij}$  for a particular attacker  $n$  is denoted as  $\mathbf{x}_n$ .

- $d_{ni}$ , 1 if attacker  $n$  targets node  $i \in T_n$ . The vector of all  $d_{ni}$  is denoted as  $\mathbf{d}$ , and the vector of all  $d_{ni}$  for a particular attacker  $n$  is denoted as  $\mathbf{d}_n$ .
- $\tilde{y}_{ij}$ , 1 if arc  $(i,j)$  is interdicted with a transparent asset; 0 otherwise.
- $\bar{y}_{ij}$ , 1 if arc  $(i,j)$  is interdicted with a trap: agents perceive the arc has not been interdicted.
- $\bar{\bar{y}}_{ij}$ , 1 if arc  $(i,j)$  is interdicted with a decoy: agents perceive the arc has been interdicted.

### Derived and artificial data

$$\tilde{r}_{nij} = \frac{\tilde{q}_{nij}}{q_{nij}} = \frac{1 - \tilde{p}_{nij}}{1 - p_{nij}}; \quad \bar{r}_{nij} = \frac{\bar{q}_{nij}}{q_{nij}} = \frac{1 - \bar{p}_{nij}}{1 - p_{nij}}; \text{ and, } \bar{\bar{r}}_{nij} = \frac{\bar{\bar{q}}_{nij}}{q_{nij}} = \frac{1 - \bar{\bar{p}}_{nij}}{1 - p_{nij}}$$

$$c_{nij} = -\log q_{nij}; \quad \tilde{c}_{nij} = -\log \tilde{r}_{nij}; \quad \bar{c}_{nij} = -\log \bar{r}_{nij}; \quad \text{and,} \quad \bar{\bar{c}}_{nij} = -\log \bar{\bar{r}}_{nij}$$

$t$ , artificial “super-sink” node

$I^* = I \cup \{t\}$  (set of nodes augmented with the super-sink node)

$A_n^* = A \cup \{(i,t) \mid i \in T_n\}$  (set of arcs augmented with arcs from destination nodes for attacker  $n$  to the super-sink node).

### Formulation

The defender tries to maximize a DVF representing expected value intercepted, which is given by:

$$\max_{\tilde{\mathbf{y}}, \bar{\mathbf{y}}, \bar{\bar{\mathbf{y}}}} \sum_{n \in N} v_n \left( 1 - \prod_{(i,j) \in A} q_{nij}^{x_{nij}} \tilde{r}_{nij}^{x_{nij} \tilde{y}_{nij}} \bar{r}_{nij}^{x_{nij} \bar{y}_{nij}} \right) \quad (1)$$

Note (1) incorporates the actual interdiction probabilities, but  $\mathbf{x}_n$  solves the flow problem for the  $n$ -th attacker, who perceives a different set of interdiction probabilities and therefore attempts to minimize the following objective:

$$\min_{\mathbf{x}_n} 1 - \prod_{(i,j) \in A} q_{nij}^{x_{nij}} \tilde{r}_{nij}^{x_{nij} \tilde{y}_{nij}} \bar{\bar{r}}_{nij}^{x_{nij} \bar{\bar{y}}_{nij}} \quad (2)$$



subject to flow balance constraints of the form:

$$\mathbf{x} \in \mathfrak{N}^* \equiv \begin{cases} \sum_{j|(i,j) \in A_n^*} x_{nij} - \sum_{j|(j,i) \in A_n^*} x_{nji} = \begin{cases} 1, & \text{if } i = s_n & (u_{n,s_n}) \\ -1, & \text{if } i = t & (u_{nt}) \\ 0, & \text{otherwise} & (u_{ni}) \end{cases}, \forall i \in I^*, \forall n \in N \\ \mathbf{x}_n \in \{0,1\}^{|\mathfrak{N}_n|}, \quad \forall n \in N. \end{cases} \quad (3)$$

Remark: The  $\mathbf{u}$ -variables refer to dual variables for the corresponding constraints.

Some manipulations to the above formulation are necessary in order to convert this bi-level problem with two different non-linear objectives into a mixed-integer problem. In order to do that, we note that:

a) We may convert the products in objective (2) into a linear objective by applying the logarithm function. This is true because  $\mathbf{x}_n^*$  solving (2) also solves:

$$\min_{\mathbf{x}_n} \log \left( \prod_{(i,j) \in A} -q_{nij}^{x_{nij}} \tilde{r}_{nij}^{x_{nij} \tilde{y}_{ij}} \bar{\bar{r}}_{nij}^{x_{nij} \bar{\bar{y}}_{ij}} \right) = \min_{\mathbf{x}_n} \sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{\bar{c}}_{nij} \bar{\bar{y}}_{ij}) x_{nij}.$$

b) If  $|N|=1$ , i.e., for the single-attacker (with a unique or multiple destinations), we may also eliminate the product in objective (1) by applying the logarithm function. This is true because the summation contains a single term,  $n = 1$ , and an optimal specification of all the  $\mathbf{y}$ -variables will also solve:

$$\max_{\tilde{\mathbf{y}}, \bar{\bar{\mathbf{y}}}} \log \left( \prod_{(i,j) \in A} -q_{nij}^{x_{nij}} \tilde{r}_{nij}^{x_{nij} \tilde{y}_{ij}} \bar{\bar{r}}_{nij}^{x_{nij} \bar{\bar{y}}_{ij}} \right) = \max_{\mathbf{y} \in Y} \sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{\bar{c}}_{nij} \bar{\bar{y}}_{ij}) x_{nij}.$$

In this case, we may solve the linear version of (1) (after taking logarithms) and then substitute back into (1) to obtain the actual expected value function sought.

c) However, if multiple attackers exist,  $|N| > 1$ , the logarithm function cannot be used directly in (1). Our assumption in this case is that (1) will still be strongly correlated to the following objective:

$$\max_{\tilde{\mathbf{y}}, \bar{\bar{\mathbf{y}}}} \sum_{n \in N} v_n \log \left( \prod_{(i,j) \in A} -q_{nij}^{x_{nij}} \tilde{r}_{nij}^{x_{nij} \tilde{y}_{ij}} \bar{\bar{r}}_{nij}^{x_{nij} \bar{\bar{y}}_{ij}} \right) = \max_{\mathbf{y} \in Y} \sum_{n \in N} v_n \left( \sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{\bar{c}}_{nij} \bar{\bar{y}}_{ij}) x_{nij} \right).$$

However, in this case we cannot claim that solutions resulting from solving this variant will yield optimal solutions to the original objective of maximizing expected value interdiction.

d) Since (1) and (2) represent different objective functions, the well-known technique of dualization of the inner problem to obtain a max-max formulation (see e.g., Brown et al. [2006]) is not applicable. However, since  $\mathbf{x}$  in (1) is defined implicitly through the constrained optimization subproblems in (2)-(3), we may use strong duality theory to replace (2) (actually, (2) as modified after taking logarithms, see remark (a)). The replacement is sets the objective of problem (2)-(3) equal to that of its dual counterpart, similarly to the technique used by Motto et al. [2005]. (Of course, we need to add additional dual constraints too.)

e) In each of the  $|N|$  subproblems in (2)-(3), binary constraints can be converted into continuous ones due to unimodularity of the shortest-path problem,  $x_{nij} \geq 0, \forall n \in N, (i, j) \in A_n^*$ . This, of course, assumes remark (a) above, where we have used the logarithm function to convert (2) into a linear-equivalent objective.

After remarks (a)-(e) are applied to the original formulation (1)-(3), the resulting formulation becomes:

$$\max_{\mathbf{y} \in Y, \mathbf{x} \in \mathbb{N}^*, \mathbf{u} \in U} \sum_{n \in N} v_n \left( \sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{c}_{nij} \bar{y}_{ij}) x_{nij} \right) \quad (4)$$

subject to:

$$\sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{c}_{nij} \bar{y}_{ij}) x_{nij} = u_{n,s_n} - u_{nt}, \quad \forall n \in N \quad (5)$$

where:

$$\mathbf{x} \in \mathbb{N}^* \equiv \begin{cases} \sum_{j|(i,j) \in A_n^*} x_{nij} - \sum_{j|(j,i) \in A_n^*} x_{nji} = \begin{cases} 1, & \text{if } i = s_n \\ -1, & \text{if } i = t \\ 0, & \text{otherwise} \end{cases}, & \forall i \in I^*, \forall n \in N \\ x_{nij} \geq 0, \forall n \in N, (i, j) \in A_n^* \end{cases} \quad (6)$$

$$\mathbf{y} \in Y \equiv \begin{cases} \sum_{(i,j) \in A} (\tilde{a}_{ij} \tilde{y}_{ij}) \leq \tilde{R} \\ \sum_{(i,j) \in A} (\bar{a}_{ij} \bar{y}_{ij}) \leq \bar{R} \\ \sum_{(i,j) \in A} (\bar{\bar{a}}_{ij} \bar{\bar{y}}_{ij}) \leq \bar{\bar{R}} \\ \tilde{y}_{ij} + \bar{y}_{ij} + \bar{\bar{y}}_{ij} \leq 1, \forall (i,j) \in A \\ \tilde{y}_{ij}, \bar{y}_{ij}, \bar{\bar{y}}_{ij} \in \{0,1\}, \forall (i,j) \in A \end{cases} \quad (7)$$

$$\mathbf{u} \in U \equiv \begin{cases} u_{ni} - u_{nj} \leq c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{\bar{c}}_{nij} \bar{\bar{y}}_{ij}, \forall n \in N, \forall (i,j) \in A \\ u_{ni} - u_{nt} \leq 0, \forall n \in N, \forall i \in T_n. \end{cases} \quad (8)$$

While all constraints  $\mathbf{y} \in Y, \mathbf{x} \in \mathbb{N}^*$ ,  $\mathbf{u} \in U$  are linear (disregarding integrality conditions on the  $\mathbf{y}$ -variables), the objective function (4) and constraint (5) are clearly non-linear, since they involve cross-products of binary and continuous variables. A linearization of these cross products can be achieved, e.g., by replacing every  $\tilde{y}x$  product by  $\tilde{x}$ , every  $\bar{y}x$  product by  $\bar{x}$ , and every  $\bar{\bar{y}}x$  product by  $\bar{\bar{x}}$ , as follows:

$$\max_{\mathbf{y} \in Y, \mathbf{x} \in \mathbb{N}^*, \mathbf{u} \in U} \sum_{n \in N} v_n \left( \sum_{(i,j) \in A} (c_{nij} x_{nij} + \tilde{c}_{nij} \tilde{x}_{nij} + \bar{\bar{c}}_{nij} \bar{\bar{x}}_{ij}) \right) \quad (9)$$

$$\sum_{(i,j) \in A} (c_{nij} + \tilde{c}_{nij} \tilde{y}_{ij} + \bar{\bar{c}}_{nij} \bar{\bar{y}}_{ij}) x_{nij} = u_{n,s_n} - u_{nt}, \quad \forall n \in N \quad (10)$$

$$\begin{aligned} 0 \leq \tilde{x}_{nij} &\leq \tilde{y}_{ij}, & \forall n \in N, \forall (i,j) \in A \\ \tilde{x}_{nij} &\leq x_{nij}, & \forall n \in N, \forall (i,j) \in A \\ \tilde{x}_{nij} &\geq \tilde{y}_{ij} + x_{nij} - 1, & \forall n \in N, \forall (i,j) \in A \\ 0 \leq \bar{x}_{nij} &\leq \bar{y}_{ij}, & \forall n \in N, \forall (i,j) \in A \\ \bar{x}_{nij} &\leq x_{nij}, & \forall n \in N, \forall (i,j) \in A \\ \bar{x}_{nij} &\geq \bar{y}_{ij} + x_{nij} - 1, & \forall n \in N, \forall (i,j) \in A \\ 0 \leq \bar{\bar{x}}_{nij} &\leq \bar{\bar{y}}_{ij}, & \forall n \in N, \forall (i,j) \in A \\ \bar{\bar{x}}_{nij} &\leq x_{nij}, & \forall n \in N, \forall (i,j) \in A \\ \bar{\bar{x}}_{nij} &\geq \bar{\bar{y}}_{ij} + x_{nij} - 1, & \forall n \in N, \forall (i,j) \in A. \end{aligned} \quad (11)$$

Thus, the IWD formulation can be finally stated as:

max (9), subject to: (6), (7), (8), (10) and (11).

The IWD is implemented in Xpress-MP (release 2007) [Dash 2007].

## C. SIMULATION

The IWD model maximizes a DVF assuming attackers perceive interdiction probabilities throughout the network in the way we expect they do, and then use them in order to optimally plan their routes to the targets. This is the premise for using decoys and traps, which disguise their actual interdiction values with perceived interdiction values. However, these assumptions need not to be the actual behavior for the attackers. Since that behavior is unknown, simulations are needed to explore how other behaviors will affect the overall effectiveness of the interdiction plan given by the optimization step.

### 1. Behaviors

We label the behavior assumed by the IWD model as a *perceptive* personality. Nominal interdiction probabilities and those of road blocks continue to be transparent for this, and all, attacker behaviors. Five other behaviors are explored: *naïve*, *communicative*, *route blocker (static)*, *route blocker (dynamic)*, and *clairvoyant*.

The *naïve* behavior assumes that the attackers have no knowledge of the network's effectiveness other than the nominal probability of interdiction on each of the arcs, which is essentially correlated to length. In other words, the attackers will try their shortest routes even if, for example, this requires that they traverse through visible road blocks indiscriminately. The simulation calculates the shortest route using Dijkstra's algorithm [Ahuja, 1993]. However, since the allocation of interdiction assets has been based on a perceptive behavior, these routes might occasionally be beneficial for the attackers.

The *communicative* behavior emulates the fact that the network's deceptive layout will degrade with time, as attackers traverse it and expose the decoys and traps. A time

horizon is provided as an additional parameter for the simulation. Throughout the time horizon, as the attackers traverse the network and get caught at an arc which is not a road block, the attackers will assume a trap and broadcast the value to next period attackers. Similarly, when an attacker reaches his target, he will broadcast that decoys traversed in his path are possible decoys and decrease their perceived effectiveness value to the nominal value.

The *route blocker (static)* behavior attempts to add an element of uncertainty to the path elected by the attackers. After an attacker calculates his path, based on the *perceptive* behavior, one arc is randomly selected and discouraged from being used by increasing its perceived probability of interdiction to 99.9%. With the modification essentially “blocking” the arc, a new shortest path is calculated and assigned to the attacker. This degrades the effect of the *perceptive* formulation and adds customization to the agent’s decision making process. This behavior emulates a pre-execution change done by the attackers. If an attacker has information about an arc which he deems as a possible delay or interdiction, then a new plan is laid which would not include that arc, unless it is the only means to reach the target. Such information could be a traffic report, a weather report, an increase in police activity, a demonstration, or any another event which renders the area untraversable.

The *route blocker (dynamic)* behavior is similar to its (static) counterpart. It adds an element of uncertainty, but this time the path changes occur during path traversal. Prior to a simulation run an arc in the path is selected at random. The attacker traverses the network based on the original path calculated with the *perceptive* behavior until it reaches the randomly selected path. When the arc is reached, we block it, and a new shortest path is calculated from the current position to the target. This adds a sub-optimal element of change to the attacker’s route. It emulates what an unforeseen change during travel will do to an attacker’s percentage of success in reaching his target.

The *clairvoyant* behavior computes the minimum interdiction probability path based on the actual interdiction values for all of the arcs. This would be the case when the defender’s plan is compromised, or if the attackers had methods of detecting the deception assets. This behavior is not expected, but it provides a worst-case value (for

the defender) utilizing deception tactics. That is, it can be used as a lower bound on the actual DVF that could be achieved by the attackers.

The following table summarizes the different behaviors and their characteristics.

Behavior	Interdiction Values Seen by Agents					
	Nominal Value	Road Blocks	Traps (Perceived)	Decoys (Perceived)	Traps (Actual)	Decoys (Actual)
Naïve	X	-	-	-	-	-
Perceptive	X	X	X	X	-	-
Communicative	X	X	X	X	Learned	Learned
Route Blocker (Static)	X	X	X	X	-	-
Route Blocker (Dynamic)	X	X	X	X	-	-
Clairvoyant	X	X	-	-	X	X

Table 6. Simulation Behaviors and Characteristics

## 2. Simulation Flow Charts

The aforementioned behaviors have been implemented in a customized ABS named the Agent Network Attack (ANA) simulation. ANA has been written in Java [Savitch 2005]. ANA uses the solution from the optimization model (described in Section B) and then simulates each attacker traversing the network arc-by-arc.

For all behaviors, except *communicative*, the overall probability of interdiction is calculated for each attacker and his generated route. The *communicative* behavior is different in that, in order to emulate communication, it has to stochastically determine the specific arc in which an attacker has been interdicted, if any. The result of these individual events determines what information is made available to all the attackers in subsequent periods.

Tallies are kept for each simulation run and for each attacker in every behavior. After all the iterations are complete for each simulated behavior a report is generated.

The following flow diagrams help explain the ANA simulation algorithms: Figure 1 shows the flow diagram for the *perceptive* behavior. Figure 2 is for the *naïve*

behavior. Figure 3 illustrates the, more complex, *communicative* behavior. Figures 4 and 5 are the diagrams for the *Route Blocker* behaviors, in their *Static* and *Dynamic* version, respectively. And, Figure 6 depicts the simulation for the *clairvoyant* behavior.

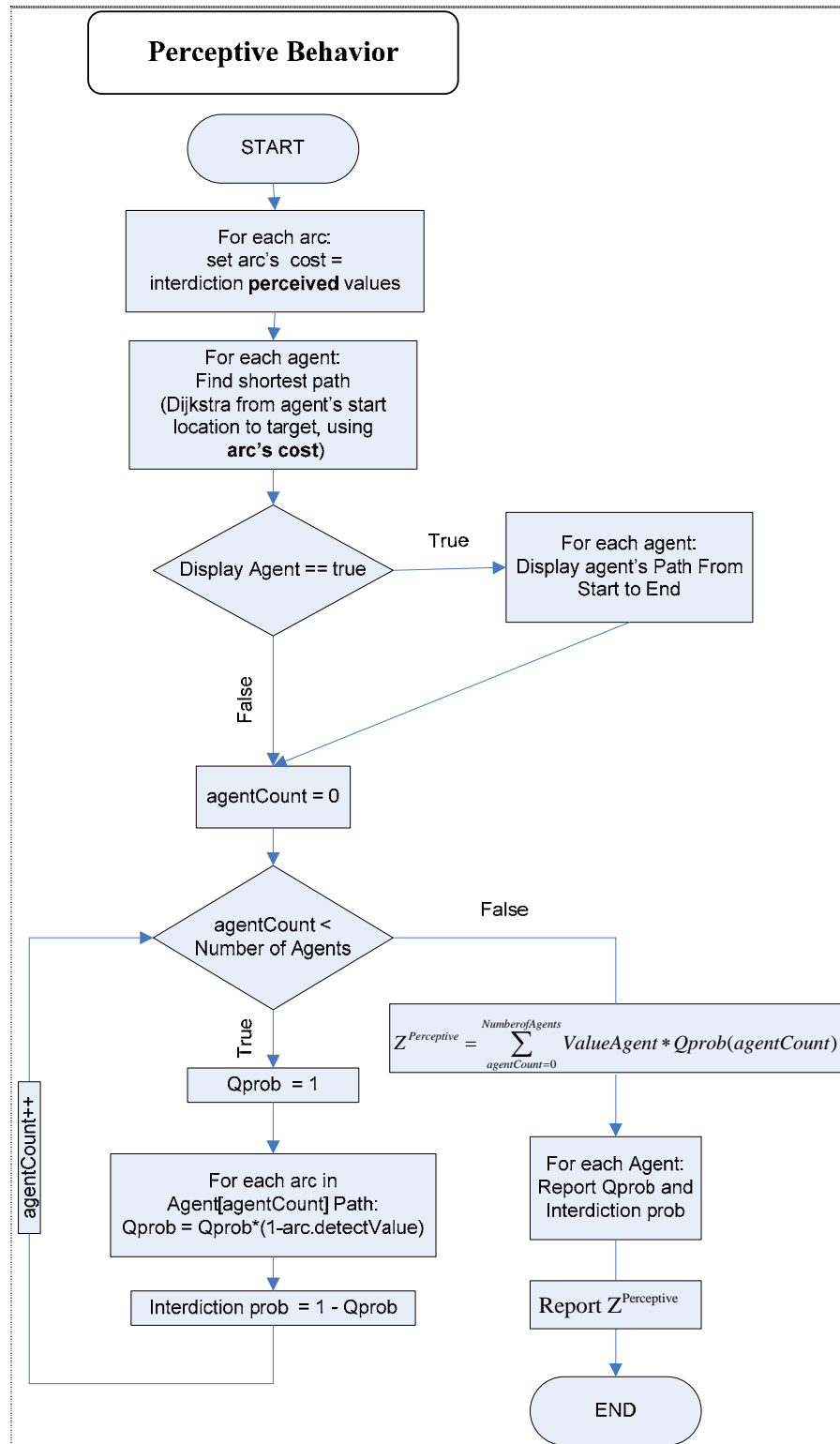


Figure 1. *Perceptive Behavior Flow Diagram*



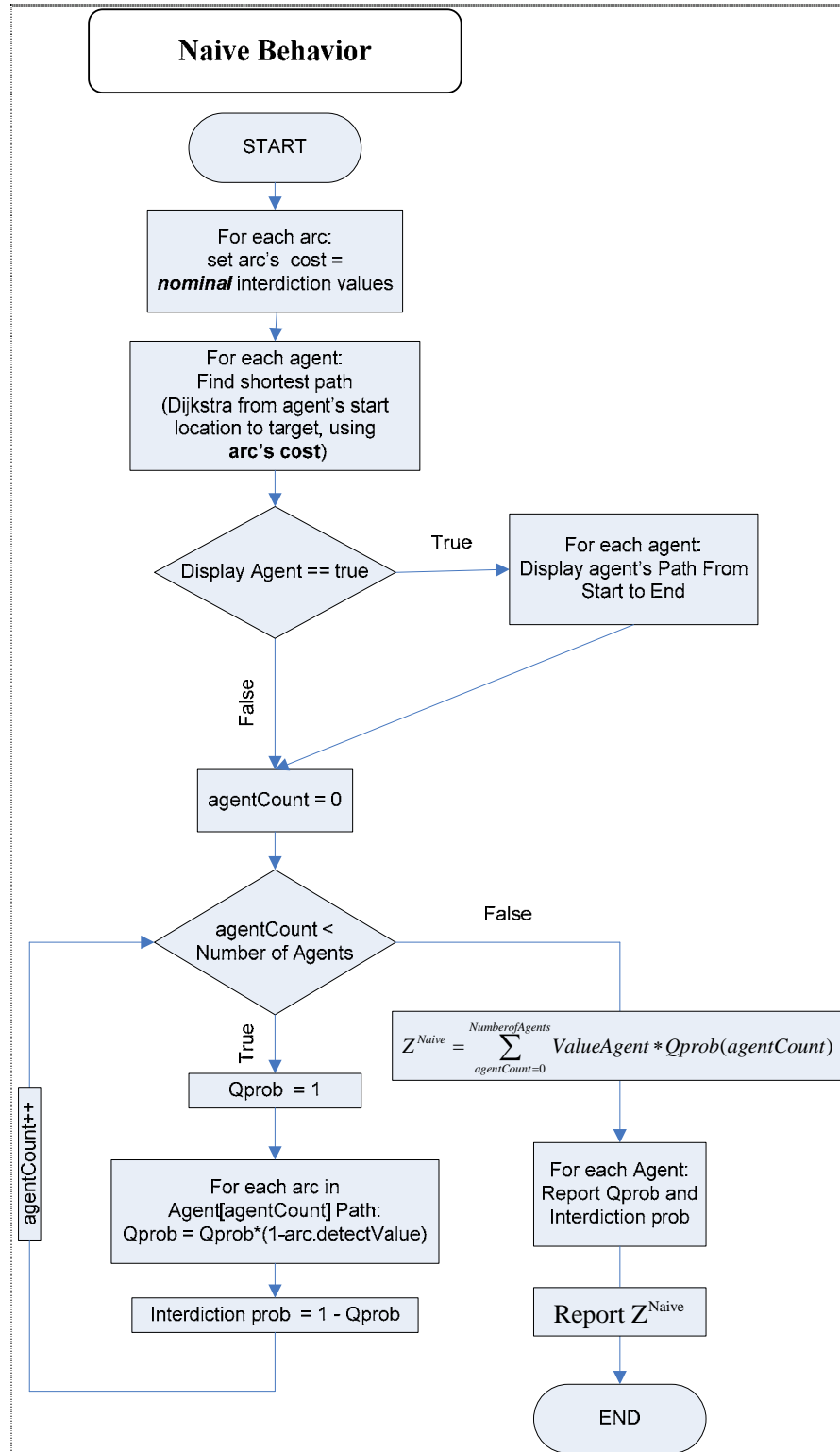


Figure 2. *Naive Behavior Flow Diagram*

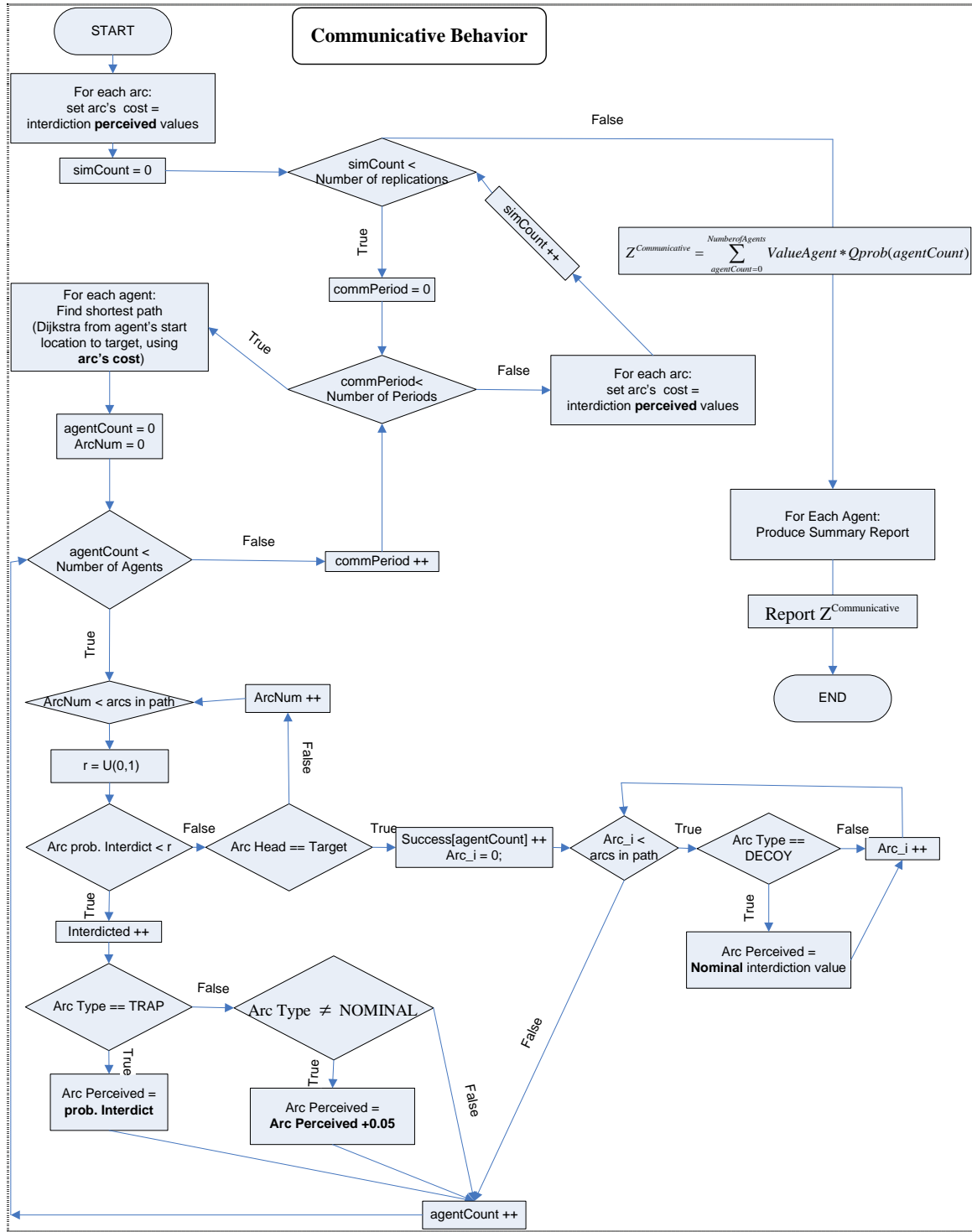


Figure 3. *Communicative Behavior Flow Diagram*

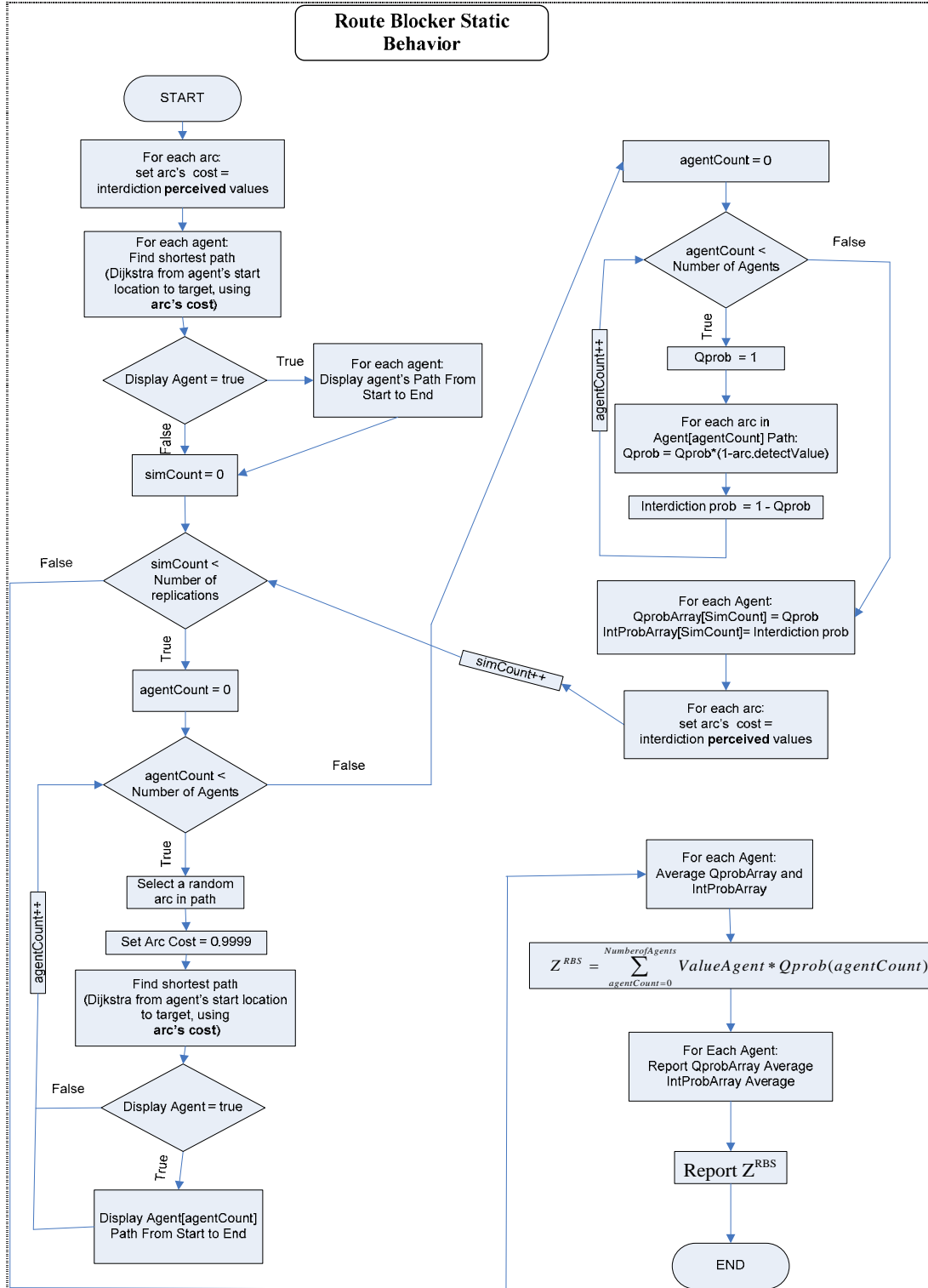


Figure 4. *Route Blocker Static Behavior* Flow Diagram

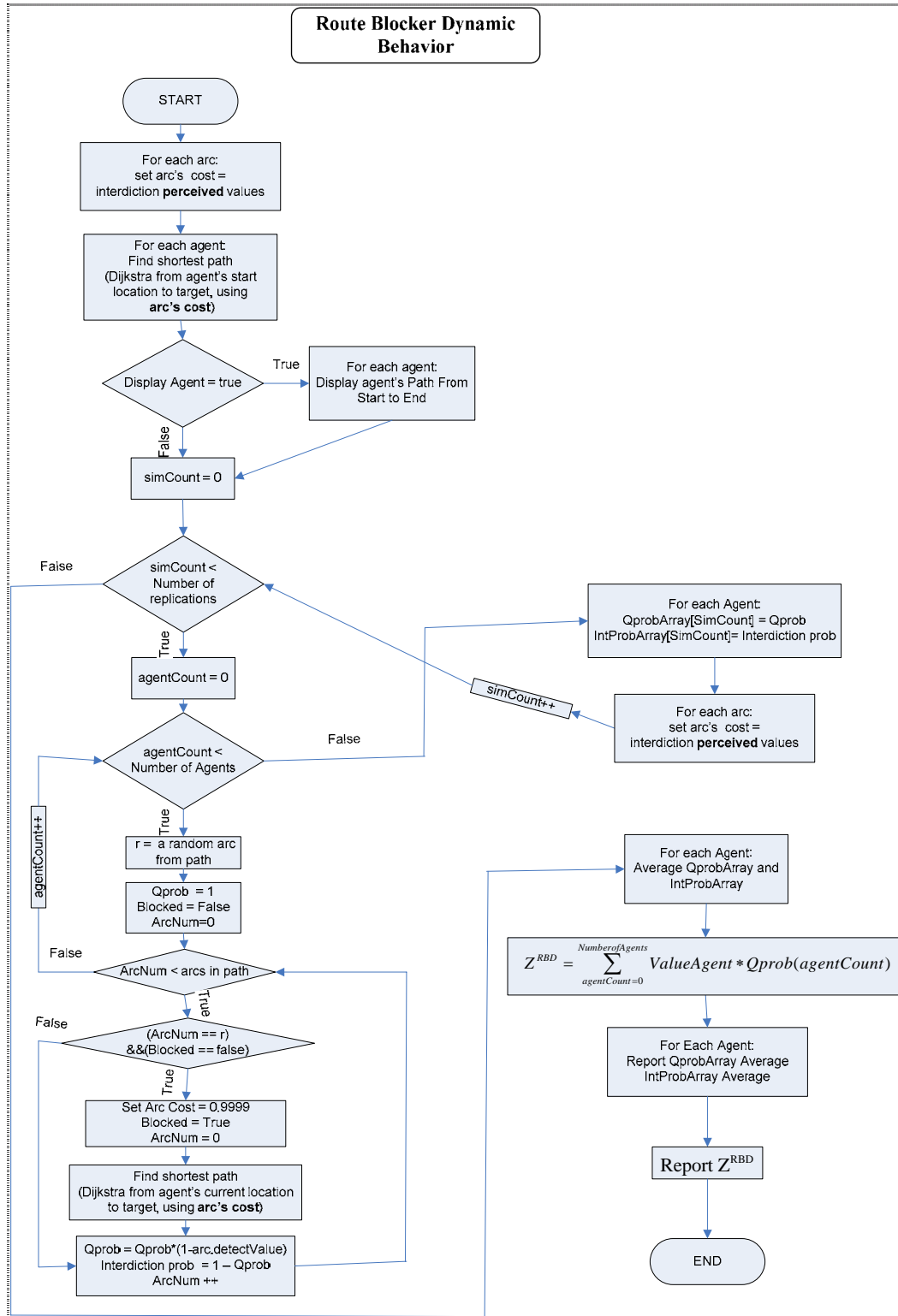


Figure 5. *Route Blocker Dynamic Behavior Flow Diagram*

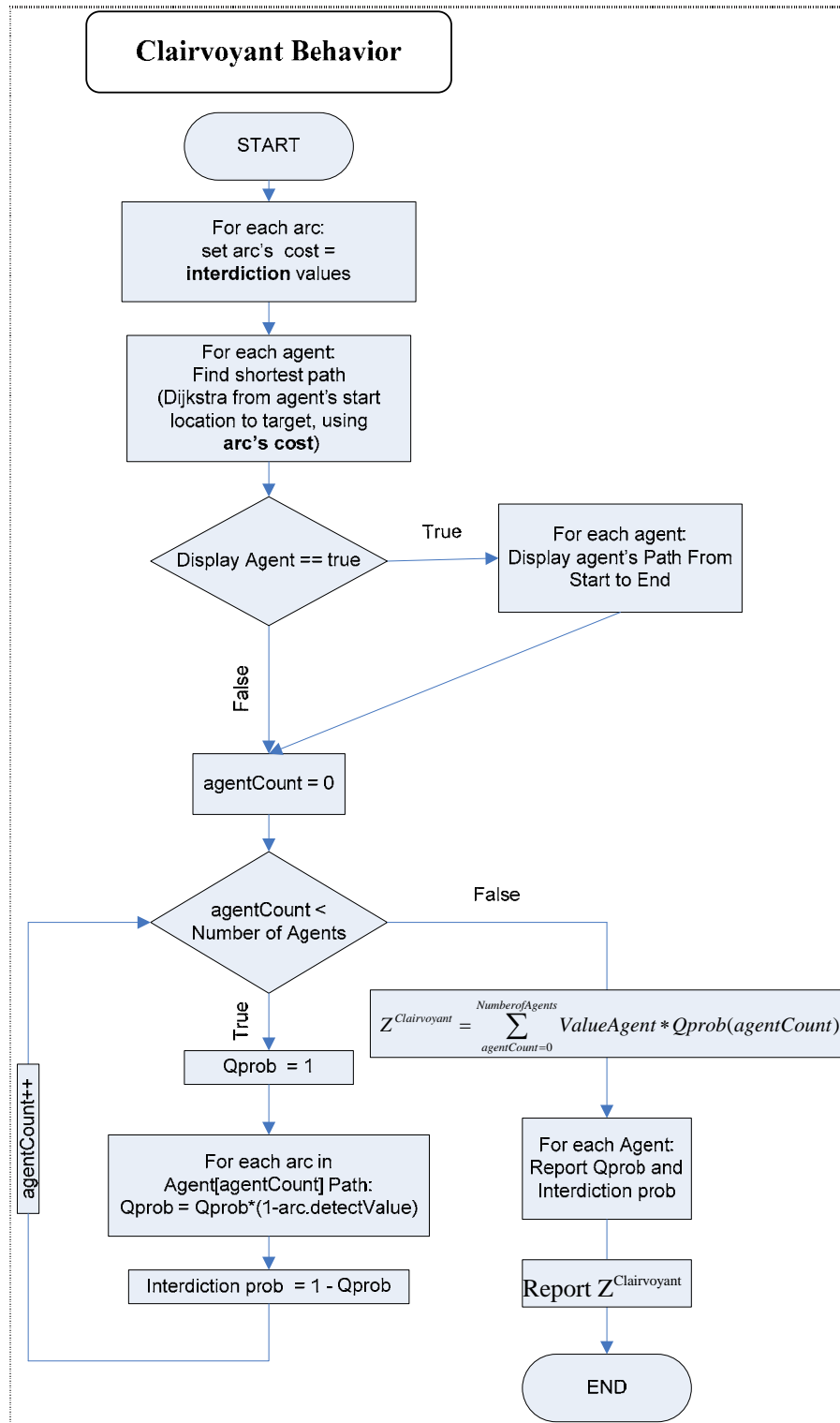


Figure 6. *Clairvoyant Behavior Flow Diagram*

THIS PAGE INTENTIONALLY LEFT BLANK

### III. COMPUTATIONAL RESULTS

This Chapter summarizes our computational experience with the IWD and the ANA simulation models. First, we describe relationships amongst different levels of interdiction resources that should apply to all test cases. Then, we illustrate the two cases we have studied, one for a small network and the other for a large network. We present the results produced by the optimization and simulation models.

#### A. SCENARIOS DESIGN AND RELATIONSHIPS

Our construct for the two cases studied consists of seven scenarios with different combinations of interdiction resources. In our computational experience, we have tried seven scenarios of interdiction resource for each test case, as presented in Table 7, where an “X” symbol indicates that the scenario uses the corresponding asset. Although we include the undefended (no interdiction) scenario for comparison purposes, we omit the scenario that contains only decoys, because in such scenario all the arcs would still be undefended. The available amount of that resource is maintained constant within the same test case.

Scenario Number	Transparent Interdictions	Traps	Decoys
I			
II	X		
III	X		X
IV	X	X	
V	X	X	X
VI		X	
VII		X	X

Table 7. General Scenario Scheme

Based on our modeling considerations and the simulated behaviors, a number of relationships and outcomes amongst behaviors and scenarios are derived. These are all

interpreted with respect to the attacker's point of view, and exclude the *clairvoyant* behavior which, for obvious reasons, is the preferred behavior regardless the scenario. Table 6 in Chapter II summarizes the behavior characteristics for the attackers.

- 1) For Scenarios I and II, the *perceptive* behavior is the best behavior to adopt by the attackers because the cases are fully transparent. Any other course of action entails the exposure to worse nominal or transparent interdiction arcs, thus increasing the probability of interdiction.
- 2) For Scenarios I and VI, the *naïve* behavior's value is equal to the *perceptive* behavior's value. In Scenario I, this is due to the absence of interdictions. In Scenario VI, it is due to the traps having no perceptive value other than nominal.
- 3) In Scenario VII, the attacker benefits from using the *naïve* behavior instead of any other behavior. This is because the decoys are ignored by the attackers and therefore the trap placement solution, which depends on the decoy's deterrence quality, is always degraded.
- 4) In Scenario III, the *naïve* behavior is better than the *perceptive* behavior. This outcome is due to the decoys being ignored, thus negating their deterrence quality and not being effective in guiding the attackers towards the transparent interdictions.
- 5) For Scenarios IV through VII, the *communicative* behavior favors the attackers when compared against the *perceptive* behavior. Any scenario that contains traps becomes less effective when these traps are exposed, which is what the *communicative* behavior does.
- 6) For Scenarios I through VII, the *Route Blocker Static* behavior is better for the attackers than the *Route Blocker Dynamic* behavior. This is because the former plans the route prior to traversing, while the latter does not change its route until it encounters a blocked arc.





replications. While this scenario is very small, applications can be found if a realistic network could be divided into smaller sub-networks, each one operating near-autonomously, or coordinated by a larger “master model.” Also, each of the above nodes could represent a larger suburb (or region), in which case the concept of interdiction on arcs needs to represent aggregated information regarding transfers between those suburbs.

The following table, Table 8, summarizes how the attackers are defined and also the values of their targets. The total target value is 60, which results from the sum of the individual attackers’ values (10, 20 and 30, respectively). This means that, if we could guarantee 100% protection of all the targets, the achieved DFV would be precisely 60.

Name	Value	targetName	SNodeName:
A1	10	3	8
A2	20	12	8
A3	30	20	8

Table 8. Basic Problem Attacker Table

## 2. Results

In the absence of interdiction assets (Scenario I) the optimal routes for the agents are through nodes 8, 4, 5, 2, 3 (for “A1”), through nodes 8, 9, 10, 11 12 (for “A2”), and through nodes 8, 14, 15, 16, 20 (for “A3”). The DVF for this scenario is 0.48, where the low value is explained by the low nominal interdiction probabilities on all arcs.

Table 9 summarizes the results for this test case. For each behavior and scenario we show estimations of total expected interdicted value,  $\hat{V}$ , and standard deviation,  $\hat{\sigma}$ . Assuming  $\hat{p}_n$  represents the point estimation for the probability of interdicting attacker  $n$  under a particular attacker’s behavior, and  $N$  is the sample size,  $\hat{V}$  and  $\hat{\sigma}$  are calculated as follows:

$$\hat{V} = v_{A1}\hat{p}_{A1} + v_{A2}\hat{p}_{A2} + v_{A3}\hat{p}_{A3} \quad (12)$$

$$\hat{\sigma} = \sqrt{v_{A1}^2 \frac{\hat{p}_{A1}(1-\hat{p}_{A1})}{N} + v_{A2}^2 \frac{\hat{p}_{A2}(1-\hat{p}_{A2})}{N} + v_{A3}^2 \frac{\hat{p}_{A3}(1-\hat{p}_{A3})}{N}}. \quad (13)$$

Note that  $V_n$  is calculated exactly ( $\sigma=0$ ) for the *perceptive*, *naïve* and *clairvoyant* behaviors because, given any defender's plan, we can calculate  $p_n$  exactly without resorting to sampling.

For the other behaviors, the Basic Problem uses  $N=5,000$ , except for the communicative behavior which employs  $N=15,000$  samples (due to the problem's horizon of three periods).

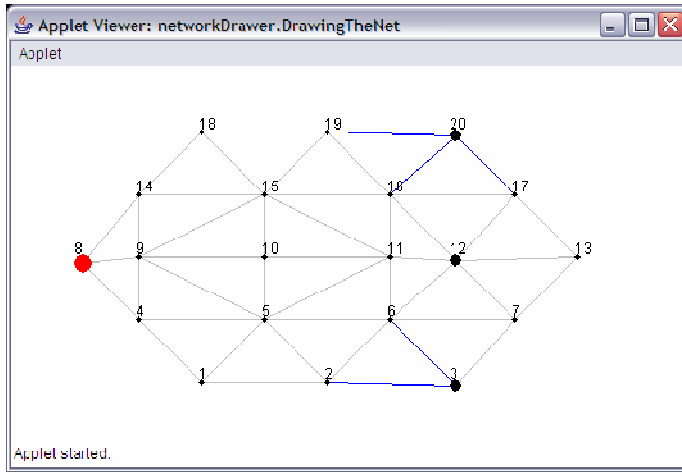
Scenario Description				Simulation Results								
Scenario	Transparent Interdictions	Traps	Decoys	Perceptive	Naïve	Communicative (3 Periods)		Route Blocker (STATIC)		Route Blocker (DYNAMIC)		Clairvoyant
				$V$	$V$	$\hat{V}$	$\hat{\sigma}$	$\hat{V}$	$\hat{\sigma}$	$\hat{V}$	$\hat{\sigma}$	$V$
Deceptive Scenarios	I			0.48	0.48	0.48	0.27	0.49	0.11	0.55	0.12	0.48
	II	5		7.88	10.33	7.95	1.89	8.30	0.21	7.95	0.21	7.88
	III	5	7	20.94	10.33	20.75	2.33	20.97	0.26	22.40	0.26	0.57
	IV	5	3	58.60	20.86	21.79	2.33	30.86	0.26	59.08	0.12	0.64
	V	5	3	59.22	41.34	33.79	2.37	42.56	0.25	59.58	0.11	0.51
	VI		3	29.77	29.77	12.61	1.99	25.45	0.24	29.91	0.12	0.49
	VII		3	59.07	10.25	33.42	2.38	39.55	0.26	59.58	0.10	0.49
				Blue's Best								
				Red's Best								

Table 9. Result Summary for Basic Problem

As expected, results are consistent with all of theoretical relationships listed in Section A. We find some case-specific outcomes, as explained in the following paragraphs.

For Scenario II, the *naïve* behavior is the worst for attackers. This scenario is fully transparent. Therefore, all other behaviors which use some perception are closer to the *perceptive* behavior, which as the theory of interdiction states is the optimal for fully transparent scenarios. Figure 8 helps illustrate how the interdictions are placed and the paths that would be selected by a *naïve* and *perceptive* behavior. (Note: The column labeled “agent” refers to “attacker” in this and subsequent figures).

## Scenario II



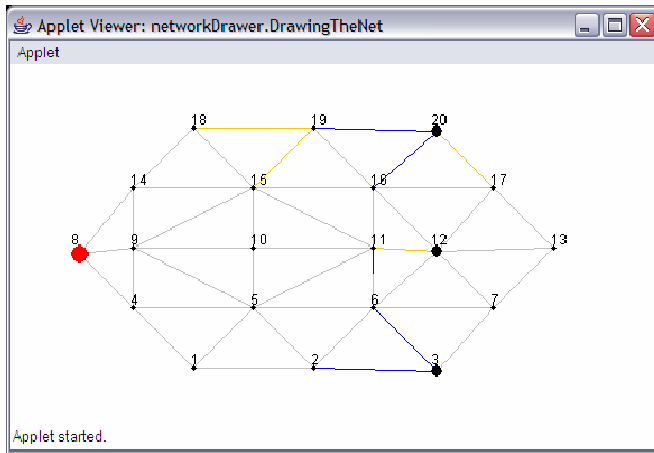
Naïve Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4	Transparent	0.0001	0.0001
	4	5		0.0030	0.0030
	5	2		0.0021	0.0021
	2	3		0.2500	0.2500
A2	8	9		0.0001	0.0001
	9	10		0.0030	0.0030
	10	11		0.0030	0.0030
	11	12		0.0016	0.0016
A3	8	14		0.0001	0.0001
	14	15		0.0030	0.0030
	15	16		0.0031	0.0031
	16	20		0.2500	0.2500
			Transparent		
			Objective Value : 10.33		

Perceptive Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4		0.0001	0.0001
	4	5		0.0030	0.0030
	5	6		0.0021	0.0021
	6	7		0.0030	0.0030
	7	3		0.2500	0.2500
A2	8	9		0.0001	0.0001
	9	10		0.0030	0.0030
	10	11		0.0030	0.0030
	11	12		0.0016	0.0016
A3	8	14		0.0001	0.0001
	14	15		0.0030	0.0030
	15	16		0.0031	0.0031
	16	20		0.2500	0.2500
			Transparent		
			Objective Value : 7.88		

Figure 8. Basic Problem, Scenario II, *Naïve* and *Perceptive* Behaviors

Scenario III is the weakest of the deceptive scenarios in terms of *perceptive* behavior for the defenders. However, since it does not contain traps, the *communicative* behavior gives nearly null advantage to the attackers. Figure 9 shows how the *naïve* behavior is advantageous to the attackers.

## Scenario III



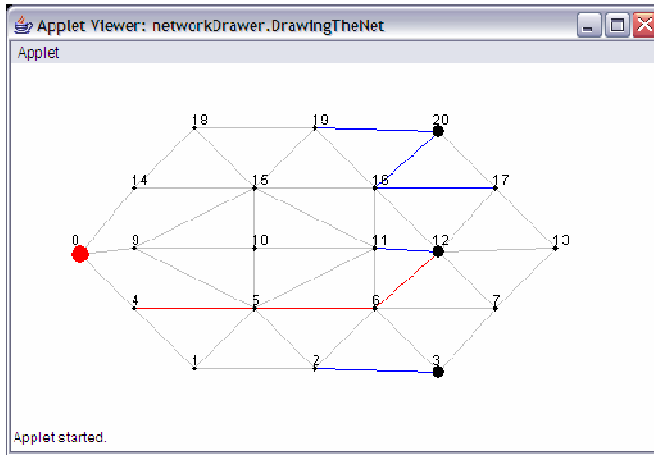
Naïve Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4	Transparent	0.0001	0.0001
	4	5		0.0030	0.0030
	5	2		0.0021	0.0021
	2	3		0.2500	0.2500
A2	8	9	Decoy	0.0001	0.0001
	9	10		0.0030	0.0030
	10	11		0.0030	0.0030
	11	12		0.2575	0.0016
A3	8	14	Decoy	0.0001	0.0001
	14	15		0.0030	0.0030
	15	16		0.2650	0.0031
	16	20		0.2500	0.2500
Objective Value :			10.33		

Perceptive Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4	Transparent	0.0001	0.0001
	4	5		0.0030	0.0030
	5	2		0.0021	0.0021
	2	3		0.2500	0.2500
A2	8	4	Transparent	0.0001	0.0001
	4	5		0.0030	0.0030
	5	2		0.0021	0.0021
	2	6		0.2500	0.2500
A3	6	12	Transparent	0.0022	0.0022
	8	4		0.0001	0.0001
	4	5		0.0030	0.0030
	5	2		0.0021	0.0021
	2	6	Transparent	0.2500	0.2500
	6	12		0.0022	0.0022
	12	16		0.0022	0.0022
	16	20		0.2500	0.2500
Objective Value :			20.94		

Figure 9. Basic Problem, Scenario III, *Naïve* and *Perceptive* Behaviors

In Scenario IV (Figure 10) we notice how the traps can be exploited very effectively by the defenders, which in turn makes the *communicative* behavior more advantageous for the attackers among perceptive-based (i.e, *non-naïve*) behaviors.

## Scenario IV



Naïve Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4	Transparent	0.0001	0.0001
	4	5		0.0250	0.0250
	5	2		0.0021	0.0021
	2	3	Decoy	0.2500	0.2500
A2	8	9		0.0001	0.0001
	9	10		0.0030	0.0030
	10	11		0.0030	0.0030
	11	12	Transparent	0.2500	0.2500
A3	8	14		0.0001	0.0001
	14	15		0.0030	0.0030
	15	16		0.0031	0.0031
	16	20	Transparent	0.2500	0.2500
			Objective Value : 20.86		

Perceptive Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	4	Trap	0.0001	0.0001
	4	5		0.0030	0.0750
	5	6		0.0021	0.0750
	6	3		0.0022	0.0022
A2	8	4	Trap	0.0001	0.0001
	4	5		0.0030	0.0750
	5	6		0.0031	0.0750
	6	12	Trap	0.0022	0.0750
A3	8	4	Trap	0.0001	0.0001
	4	5		0.0030	0.0750
	5	6		0.0031	0.0750
	6	12	Trap	0.0022	0.0750
	12	17		0.0022	0.0022
	17	20		0.0021	0.0021
			Objective Value : 58.6		

Figure 10. Basic Problem, Scenario IV, *Naïve* and *Perceptive* Behaviors

As expected, given that it contains more resources, Scenario V (shown in Figure 11) is the best scenario for the defenders, regardless of the simulated behavior. (Note: Also expected per Section B, the *communicative* behavior is best for the attackers.)

Applet Viewer: networkDrawer.DrawingTheNet

Applet

Applet started.

Perceptual Behavior					
Agent	From	To	Interdiction Type	Perceived Value	Real Value
A1	8	14		0.0001	0.0001
	14	15	Trap	0.0030	0.7500
	15	11	Trap	0.0021	0.7500
	11	12	Trap	0.0016	0.7500
	12	7		0.0021	0.0021
	7	3	Trans parent	0.2500	0.2500
A2	8	14		0.0001	0.0001
	14	15	Trap	0.0030	0.7500
	15	11	Trap	0.0021	0.7500
	11	12	Trap	0.0016	0.7500
A3	8	14		0.0001	0.0001
	14	15	Trap	0.0030	0.7500
	15	11	Trap	0.0021	0.7500
	11	12	Trap	0.0016	0.7500
	12	13		0.0030	0.0030
	13	17	Trans parent	0.2500	0.2500
	17	20		0.0021	0.0021
Objective Value :			59.22		

Figure 12 illustrates Scenario VI. We observe that all of the traps are placed on arcs corresponding to attacker “A3,” achieving a DVF of 30 (almost 50%). This solution is clearly suboptimal because one trap could have been placed on each attacker’s path, which would have yielded a DVF of at least 45 (75%). This is a consequence of the objective function used by the IWD, which correlates but does not fully represent the concept of expected value in the case of multiple attackers. In general, the *communicative* behavior helps the attackers the most amongst all the scenarios that contain traps. This is even more emphasized when the IWD solution is suboptimal as indicated for this scenario: The traps have been placed so that the interdiction of the

highest-value target is maximized, affecting attacker “A3” only; as soon as they are exposed by the *communicative* behavior, new routes direct the attacker away from arcs with those traps.

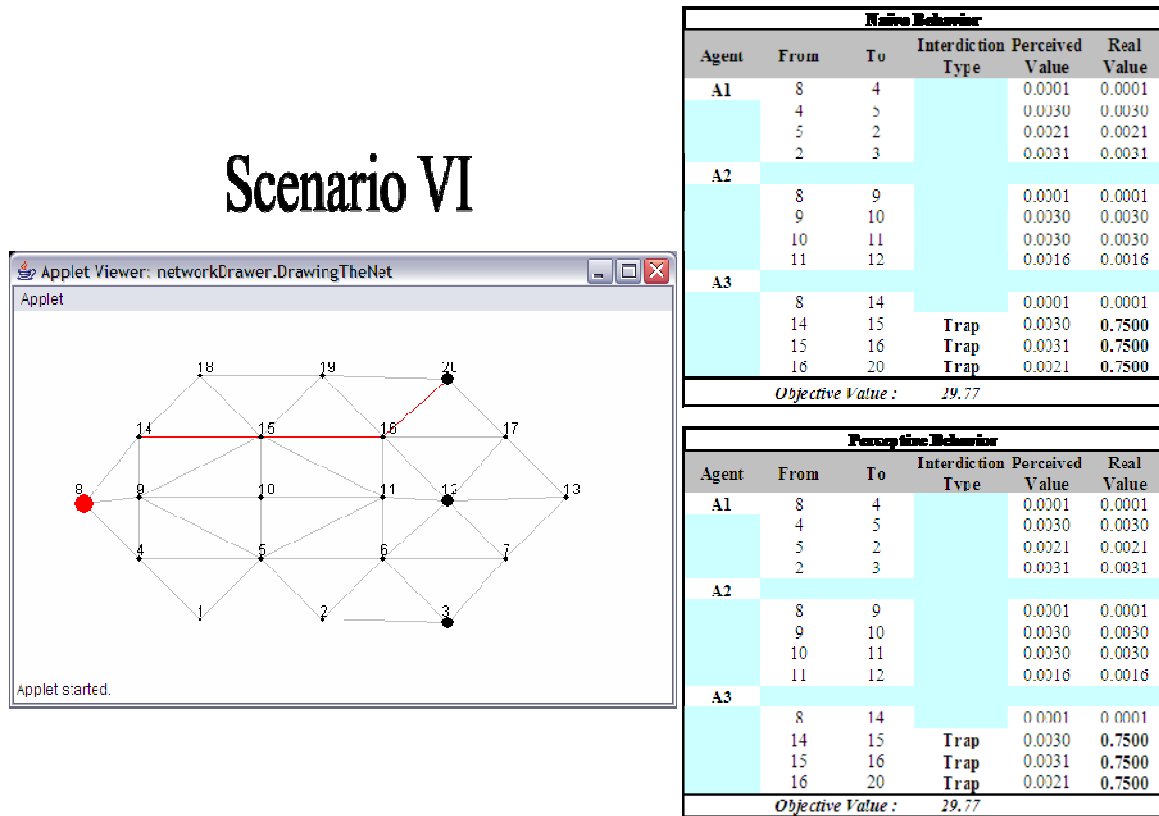
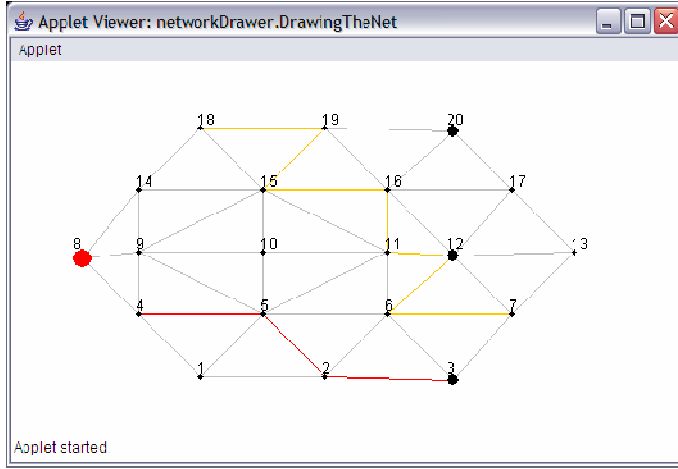


Figure 12. Basic Problem, Scenario VI, *Naïve* and *Perceptive* Behaviors

Scenario VII demonstrates how a *naïve* behavior may defeat the *perceptive* behavior assumed to place the interdiction assets. All attackers would be deceived by decoys to take routes that contain traps, if they acted with a *perceptive* behavior. However, the *naïve* behavior leads to routes where only one low-value attacker traverses a trap.



## Scenario VII



Naïve Behavior					
Agent	from	to	Interdiction Type	Perceived Value	Real Value
A1	8	4		0.0001	0.0001
	4	5	Trap	0.0030	0.7500
	5	2	Trap	0.0021	0.7500
	2	3	Trap	0.0031	0.7500
A2	8	9		0.0001	0.0001
	9	10		0.0030	0.0030
	10	11		0.0030	0.0030
	11	12		0.0016	0.0016
A3	8	14		0.0001	0.0001
	14	15		0.0030	0.0030
	15	16		0.0031	0.0031
	16	20		0.0021	0.0021
Objective Value:				10.25	

Perceptive Behavior					
Agent	from	to	Interdiction Type	Perceived Value	Real Value
A1	8	4		0.0001	0.0001
	4	5	Trap	0.0030	0.7500
	5	2	Trap	0.0021	0.7500
	2	3	Trap	0.0031	0.7500
A2	8	4		0.0001	0.0001
	4	5	Trap	0.0030	0.7500
	5	2	Trap	0.0021	0.7500
	2	3	Trap	0.0031	0.7500
	3	7		0.0021	0.0021
	7	12		0.0022	0.0022
	12	17		0.0021	0.0021
A3	8	4		0.0001	0.0001
	4	5	Trap	0.0030	0.7500
	5	2	Trap	0.0021	0.7500
	2	3	Trap	0.0031	0.7500
	3	7		0.0021	0.0021
Objective Value:				59.07	

Figure 13. Basic Problem, Scenario VII, *Naïve* and *Perceptive* Behaviors

All the above experiments have used three periods for the *communicative* behavior. Since the number of communication periods affects the DVF in this behavior, we have performed some experiments with five and seven communication periods. Table 10 and Figure 14 show the results for this experiment, where the impact of longer communication periods is realized in those scenarios involving traps (Scenarios IV through VII).

Scenario Description				Communicative Simulation Results			
Scenario	Number Interdictions	Number Traps	Number Decoys	Perceptive Behavior	Communication Periods		
					3	5	7
I				0.5	0.48	0.49	0.49
II	5			7.9	7.95	7.87	7.86
III	5		7	20.9	20.75	20.73	20.88
IV	5	3		58.6	21.79	13.38	9.82
V	5	3	7	59.2	33.79	24.85	20.72
VI		3		29.8	12.61	8.07	5.93
VII		3	7	59.1	33.42	20.41	14.78

Table 10. Communicative Behavior Results for three Communication Periods

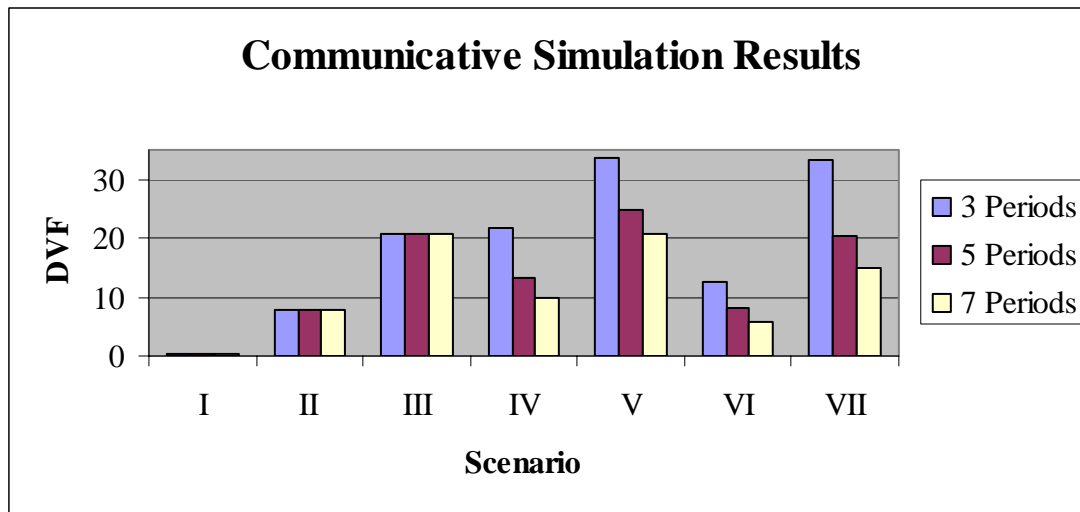


Figure 14. Communicative Behavior Results for Different Periods per Scenario

In a scenario where no deception is utilized (Scenarios I and II), the communicative behavior hinders the attackers. This is due to the effect of nominal interdictions: If a nominal interdiction occurs, then an agent is forced to believe that a trap exists in that arc and therefore alter its path and increase its risk of interdiction by transiting more (or less attractive) nominal arcs. A similar situation occurs for Scenario III where only decoys are deceptive. Here, since decoys are seldom traversed (because of their nature as decoys), attackers also tend to traverse nominal arcs or transparent interdictions only, as in Scenarios I and II.

This kind of experiment can help planners in determining how frequent to change the placement of interdiction assets in order to mitigate the effects of communication amongst attackers.

## C. DYSTOPIA

### 1. Description

Dystopia (Figure 15 [Locke, 2007]) is a notional region that consists of two cities, Grim City and Cape Hazard. Dystopia was developed as part of a homeland security initiative which required a city to serve as a test case while keeping any results of an investigation at the unclassified level. Figure 16 shows Dystopia's road network. Cape Hazard's road system has been chosen as our case study. Figure 17 shows a closer view of Cape Hazard's road network.

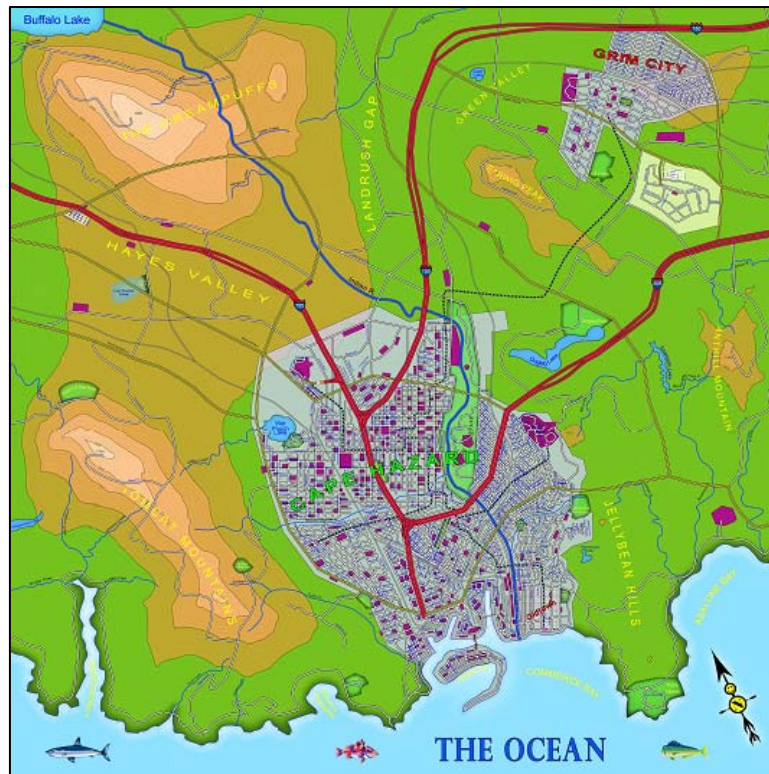


Figure 15. Dystopia

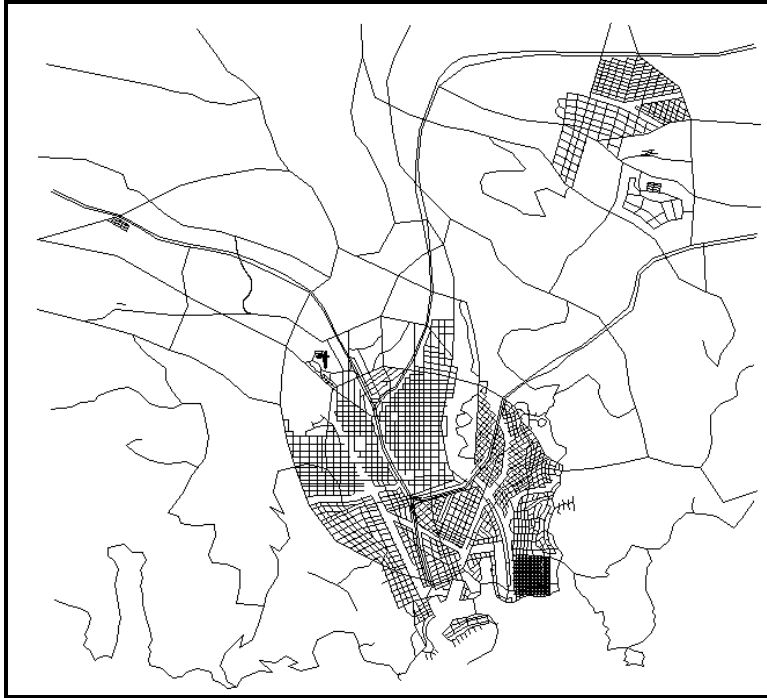


Figure 16. Dystopia in Shape File Format

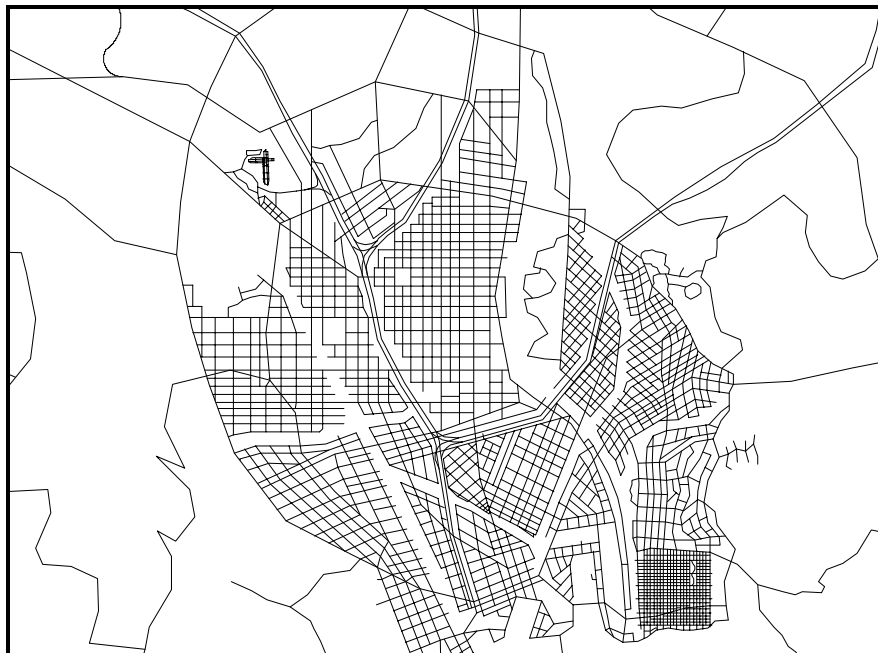


Figure 17. Cape Hazard in Shape File Format

The data for Dystopia comes in the form of shape files. A technical issue arises when dealing with shape files. Shape files are designed for compact and common

geographical data exchange; however, as is, the data contained in the file does not lend itself for network analysis: it requires conversion into an appropriate node and arc construct with which to build the network structure used by our optimization and simulation models. To overcome this difficulty, we have developed a tool which performs this non-trivial task. Details are included in Chapter IV.

By using the converting tool, we transform Dystopia into a network consisting of 4,922 nodes and 6,927 arcs (see Figure 18). Unfortunately, the conversion has some side effects. For example, many of these arcs will correspond to the same road segment when the original road has curvature. This increases the problem size, which in turn hinders our ability to achieve optimal solutions to the IWD optimization model, and limits the number of replications our ANA simulation can carry out in affordable time.

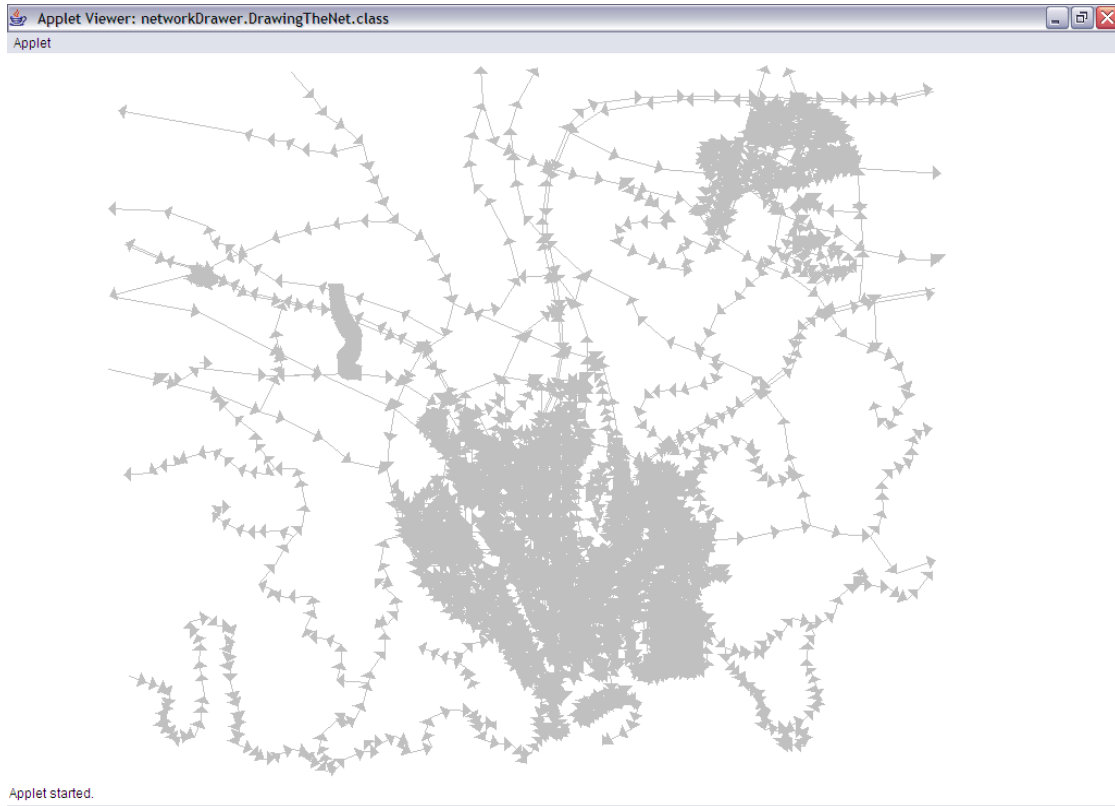


Figure 18. Dystopia Converted into Nodes and Arcs

Figure 19 displays the network for the underlying case with starting locations and targets for three attackers. The starting locations are near the city limits. The target for

each attacker has been designated based on proximity to the starting position. In particular, two transportation hubs and one police station have been chosen for this case (see Table 11). The total value of these targets is 200.

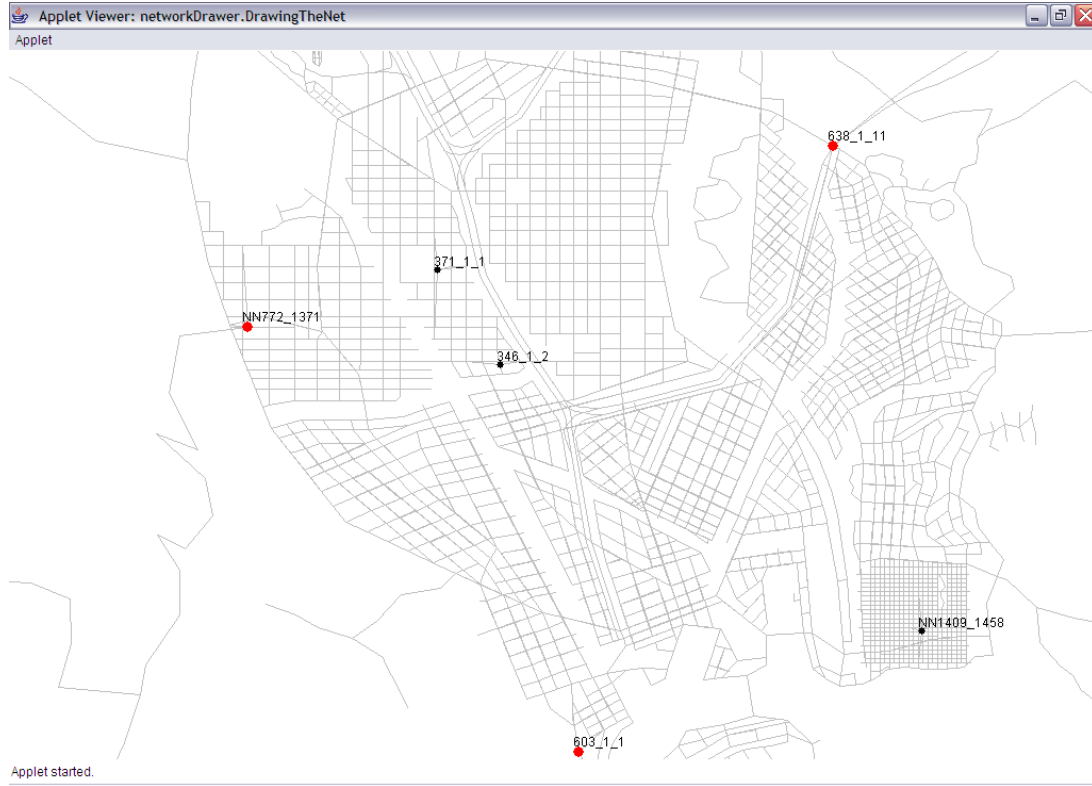


Figure 19. Cape Hazard with Attackers and Targets

Name	Value	Start Node Name	Target Name	Target Description
A1	100	603_1_1	NN1409_1458	Old Towne Train Stop
A2	60	NN772_1371	371_1_1	Grey Hound Bus Terminal
A3	40	638_1_11	346_1_2	Police Station (Marlboro Station)

Table 11. Dystopia Scenario Agent Table

For this test case, and given the size of the road network, we assume a maximum of 50 transparent interdictions, 25 traps and 75 decoys, combined in different ways depending on the scenario.

## 2. Results

Table 12 shows the summary of the results where estimated  $\hat{V}$  and  $\hat{\sigma}$  are calculated as in equations (12) and (13). Like in the Basic Problem, the values for perceptive, *naïve* and *clairvoyant* behaviors are calculated exactly. We employ  $N = 500$ , except for the communicative behavior which uses  $N = 3,500$ .

We first caution the reader that the IWD solutions obtained for Scenario V and Scenario VII have not converged to their optimal values. The optimality gap after eight hours of execution for these two scenarios is over 100%. We use the solution for Scenario IV as a surrogate feasible solution for Scenario V. Likewise, we use the solution for Scenario VI with Scenario VII.

As with the Basic test case, the results conform with theoretical relationships listed in Section A. Also, the *communicative* behavior improves the attackers' performance for scenarios where traps are utilized (Scenarios IV through VII). Also, as in that case, Scenario V yields the best or near-best strategy for the defender. However, since we have used the optimal solution to Scenario IV as a substitute for Scenario V, we cannot ascertain the gain of using decoys as we did in the Basic test case.

The Appendix contains graphical illustration of the of the interdiction plans.

Scenario Description				Simulation Results									
Scenario	Transparent Interdictions	Traps	Decoys	Perceptive	Naïve	Communicative (7 Periods)		Route Blocker (STATIC)		Route Blocker (DYNAMIC)		Clairvoyant	
				$V$	$V$	$\hat{V}$	$\hat{\sigma}$	$\hat{V}$	$\hat{\sigma}$	$\hat{V}$	$\hat{\sigma}$	$V$	
Deceptive Scenarios	I			0.94	0.94	1.13	4.31	1.11	0.33	1.34	0.35	0.94	
	II	50		158.16	158.16	158.61	10.76	158.17	0.65	171.21	0.56	158.14	
	III	50		75	181.29	62.83	181.53	8.76	178.82	0.55	191.35	0.41	33.60
	IV	50	25		196.01	196.01	181.39	7.36	193.92	0.34	198.74	0.17	158.25
	V *	50	25	75	196.01	196.01	181.39	7.36	193.92	0.34	198.74	0.17	158.25
	VI		25		160.27	160.27	109.50	8.81	159.30	0.25	160.38	0.13	1.71
	VII *		25	75	160.27	160.27	109.50	8.81	159.30	0.25	160.38	0.13	1.71
					Blue's Best								
				Red's Best									
* IWD Not Solved Optimally													

Table 12. Result Summary for Dystopia Scenarios

THIS PAGE INTENTIONALLY LEFT BLANK



## **IV. GRAPHICAL USER INTERFACE**

A Graphical User Interface (GUI) application has been built which allows for the creation and execution of the network interdiction optimization and simulation models described in Chapter II. This chapter provides an overview of the GUI and other tools built and/or used in conjunction with the GUI. As an integrated application, the GUI is intended to become a user-friendly environment which aids with problem data preparation and validation, setting up and running the optimization and simulation models, and interpreting the results through visualization features, which include a graphical depiction of the problem network. The reader and potential user of the GUI is cautioned that, at the time of this writing, the GUI is still in an early, prototypic stage.

### **A. DESIGN**

#### **1. Tables, Queries and Macros**

The GUI has been built in MS-Access [Microsoft®, 2003] and contains a set of queries, forms, and helper tables to aid the user in managing the problem data, that is, input, deletion, modification and general manipulation of all the sets and parameters delineated in Chapter II. The GUI also contains macros which allow the user launching the IWD optimization model and the ANA simulation model, and retrieve their results.

Figure 20 lists the tables used by the GUI. All of these tables are linked to two independent databases which contain specific tables for the IWD and ANA models. In particular, tables with names ending in “(Xpress)” are linked to the IWD optimization model. For example, table Arc(Xpress) contains exclusively the arc information used by the IWD model, namely the arc tail and head nodes, the interdictability character of the arc, and the values for nominal, transparent, trap and decoy probabilities of detection on the arc. On the other hand, tables ArcList and ArcTypes contain specific ANA data, such as the arc index values for array referencing, and the arc type nominal multiplier values for nominal interdiction value calculations.

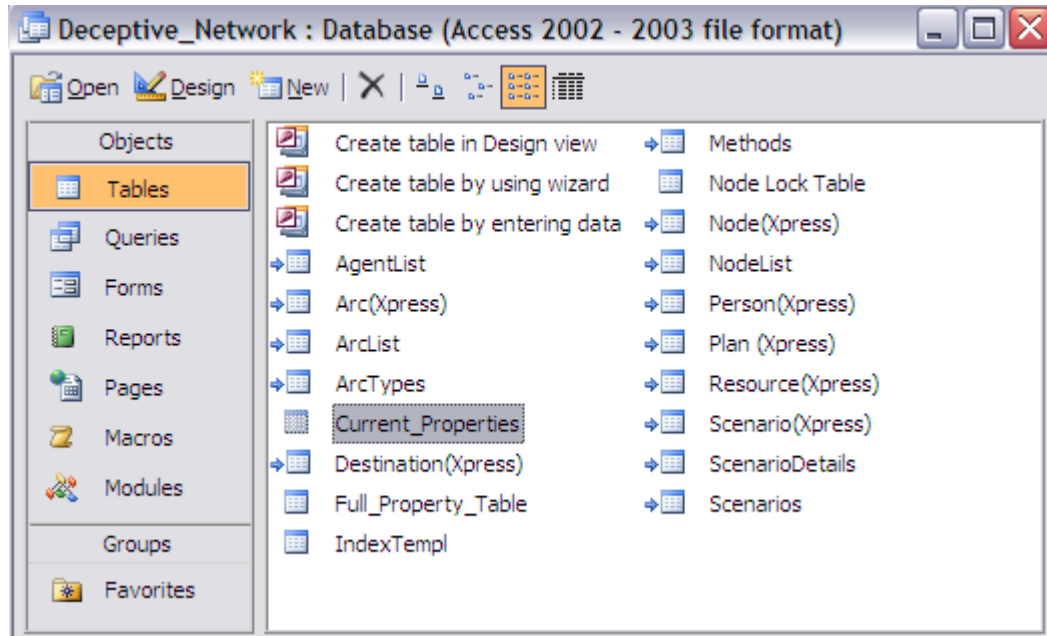


Figure 20. GUI: Supporting Table List

Figure 21 shows the list of queries built for the GUI. These range from “output” (consult) queries, which export the data for the simulation, to “update” queries which calculate and modify the detection values for the individual arcs according to the methods described in Chapter II. For example, output query *ArcOut* selects all the arcs that define the network, its individual interdiction values and the optimization solution and formats these data for exporting as text into a file called *arclist.txt*. The update query *update\_Interdiction\_Type* takes the solution from the *Plan(Xpress)* table and translates it into text values to place on the *ArcList*’s *Interdiction\_Type* field. This allows the ANA simulation to obtain the placements of the interdiction assets.

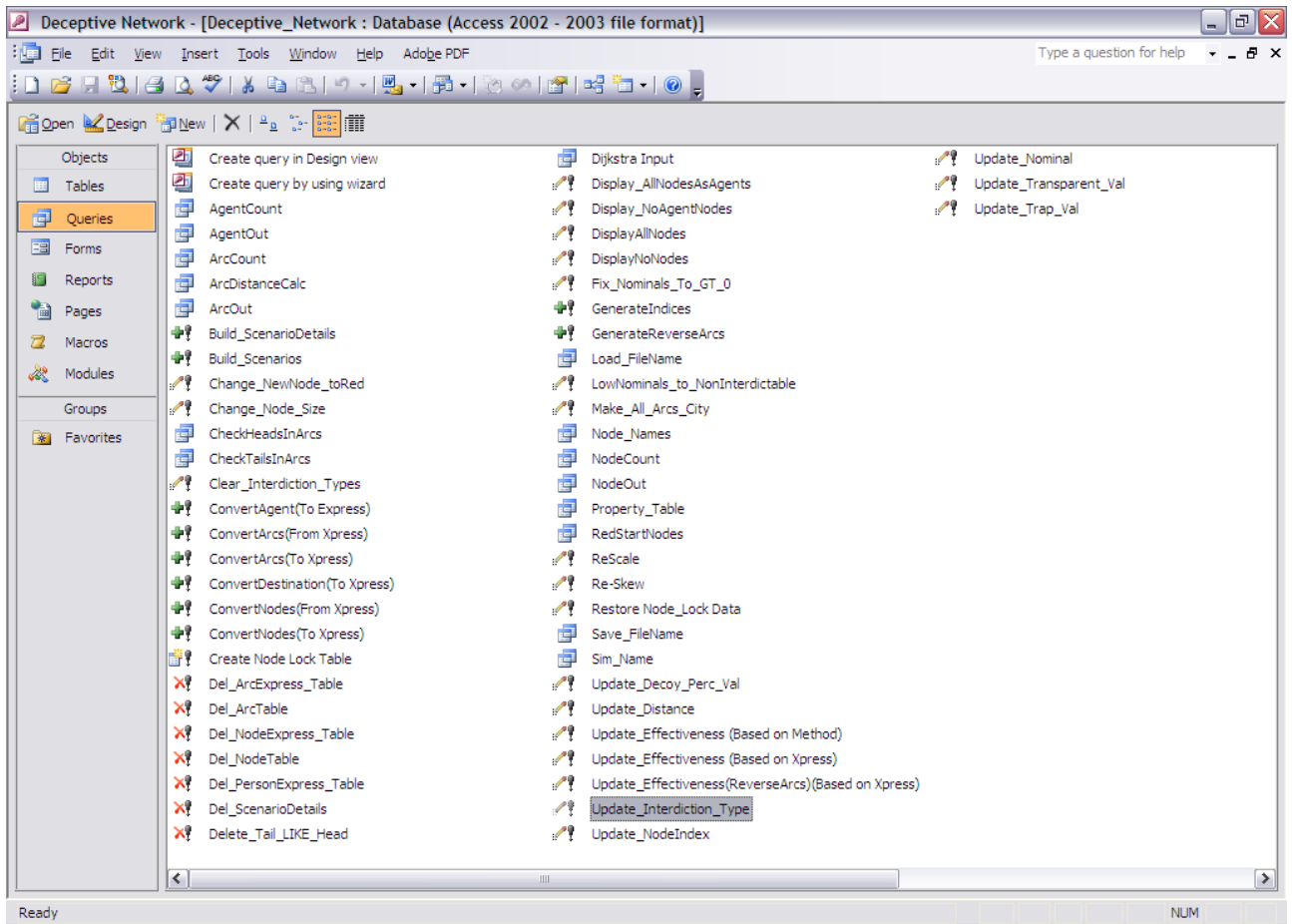


Figure 21. GUI: Supporting Query List

Figure 22 lists the “macros” contained in the interface. Each macro consists of sets of instructions to perform. For example, the *Export Network* macro updates all necessary array indices, and exports all of the text files required for the ANA simulation.

Both queries and macros can be activated manually by selecting them from the lists, or initiated via buttons on forms, which are described in detail in the remainder of this section.

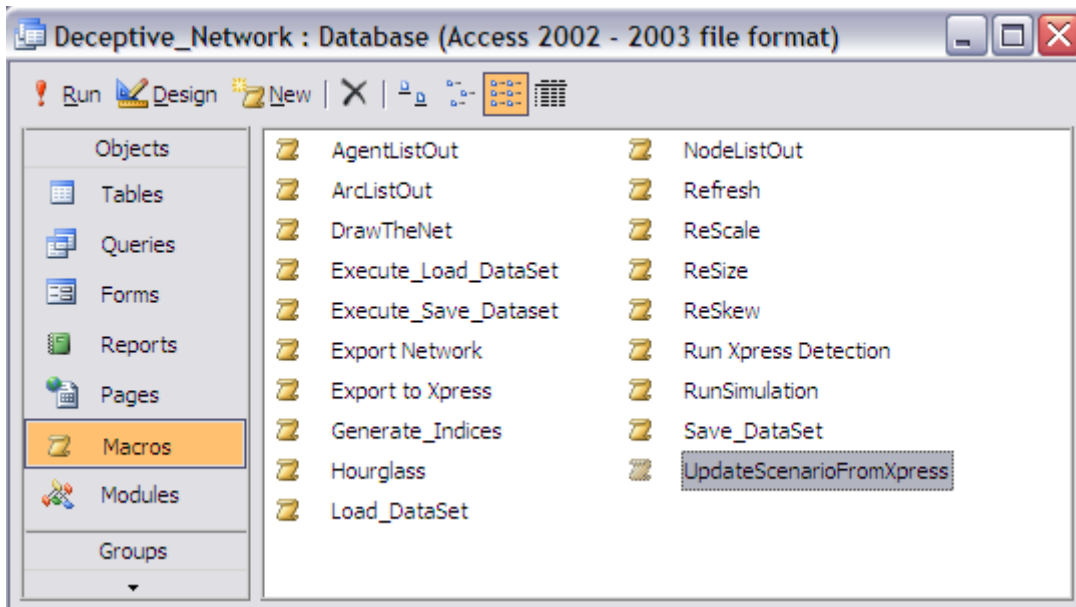


Figure 22. GUI: Supporting Macro List

## 2. Forms

Forms help display the data contained in the tables and queries. They contain buttons that execute queries or macros. We next describe how the forms interact with the data tables and the queries. The forms for our GUI are listed in Figure 23.

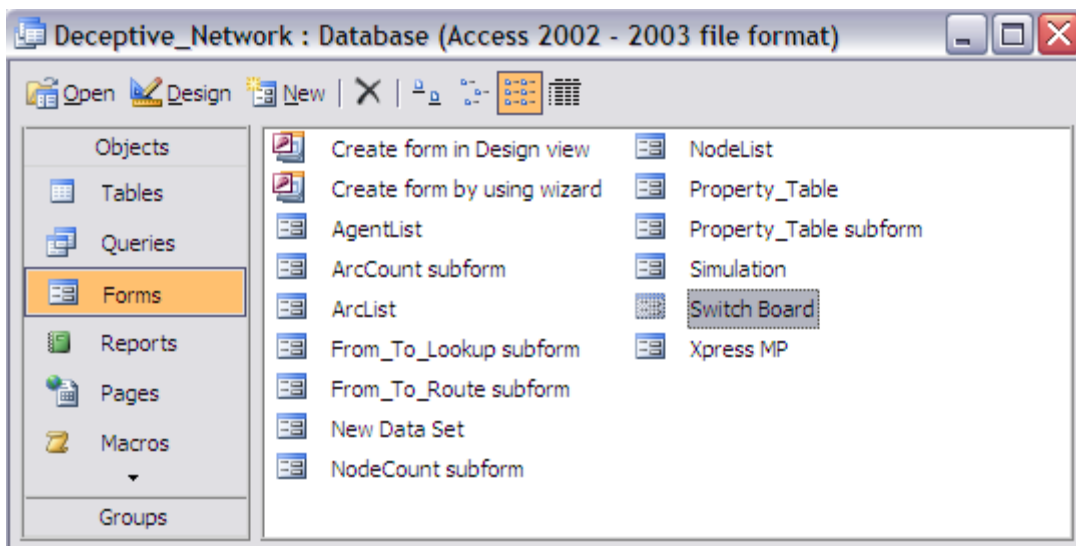


Figure 23. GUI: Supporting Form List

When a user opens the GUI the Switch Board Form (Figure 24) opens by default. The Switch Board Form is the so-called “main” form for the GUI. Network editing functions, model parameters and simulation control functions are accessed via this form. We now describe all the features that can be accessed through buttons and other objects on the Switch Board Form.

Switch Board: “File” multi-tab:

File control is performed on the Switch Board Form via the “File” multi-tab object: The “Data Set” tab allows the user to load an existing problem or to save the incumbent one, whereas the “New Data Set” tab allows the user to create a brand new case from scratch or from the current one loaded.



Figure 24. Switchboard Form

Switch Board: “Network” button:

The “Network” button gives access to the “Forward Star” form (Figure 25). This is a main editing form for the network data. It allows control of all the data elements shown in the figure.

The top section of the “Forward Star” form allows for Node data elements to be controlled, while the bottom sub-form allows for editing the arc attributes.

Editable data for a node are its name and long name, and X-Y location. Other attributes such as size and color are used for graphical visualization only. Arc data includes its type (e.g., city street or highway), length, and detection probabilities based on the type of interdiction (if any) or otherwise nominal.

This form also shows the IWD optimization model solution under the column labeled “interdiction\_type,” which refers to the type of interdiction asset placed on the arc, if any.

In order to transfer the data to the simulation module, certain queries need to be executed. The “Export Network” button processes the data into the required format and exports all the necessary data files.

The “Network Drawer” button displays the problem network graphically, as described below.

Finally, the “Calculate Nominal Values” button executes the *update\_Nominal* query, which in turn updates the nominal values based on the arc lengths.

Remark: Although this form can be used to input a network from scratch, this capability is not intended to be used with large networks. If the data is contained in another database or in a spreadsheet, a user can utilize the MS-Access import capabilities to feed the data to the GUI (as long as certain conditions are met, such as field name matching in both data storages). The next section in this chapter addresses a method for generating a network which illustrates this.

Node: 346\_1\_2 Location\_X: 457.7440390 Location\_Y: 290.5519367 Long\_Name: Police Station Marlboro Circle [Delete Node]

Size: 7 Color: BLACK [v] ☒ Display\_Node\_Name

Notes: This Police Station handles recruiting for the city and in difficult times has lines of recruits outside the staion.

[Export Network] [NETWORK Drawer] [Calculate Nominal Values]

☐ Use Distance as Cost

Total Nodes: 4922 Total Arcs: 6927

	Head	Distance	Cost	Type	Nominal	Transparent	Trap	Decoy_Perc	Interdiction_Ty
▶	345_1_1	22	0	City	0.0002	0.25	0.75	0.25105	
	NN691_767	13	0	City	0.0001	0.25	0.75	0.2506	
	NN692_714	10	0	City	0.0001	0.25	0.75	0.25045	
*									

Record: 813 of 4922

Figure 25. Forward Star Form for Network Data Input

Switch Board: “Property Table” button:

The “Property Table” form (Figure 26), allows for batch changes to the network nodes as well as X and Y transformation in order to aid the user obtain a better graphical display of the network. For example, the user may modify the X-Y scale and/or skew factors in order to see more detail on a dense network. In addition, the user may opt to display a subset of nodes, attackers, etc.

Of course, the elements in this form do not alter the model parameters or its solutions.



The screenshot shows a window titled "Property\_Table" with a standard Windows-style title bar (minimize, maximize, close buttons). The form contains several input fields and buttons:

- Use\_Distance**: A checkbox that is currently unchecked.
- File\_NickName**: A text field containing "Dystopia".
- FileName**: A text field containing "Dystopia.mdb".
- Default\_ArcType**: A dropdown menu showing "City".
- Default\_NodeValue**: A text field containing "1".
- X\_Scale\_Factor**: A text field containing "1.1".
- Y\_Scale\_Factor**: A text field containing "1.1".
- Default\_Node\_Size**: A text field containing "0".
- Buttons**:
  - Resize Nodes**: A button.
  - ReScale**: A button.
  - Display all Node Names**: A button.
  - Display NO Node Names**: A button.
  - All Nodes ARE Agents**: A button.
  - NO Agent Nodes**: A button.
  - EXPORT NETWORK**: A button.
  - Create Restore Point**: A button.
  - Restore Node Positions**: A button.
- Skew Controls**:
  - X\_Skew**: A text field containing "0".
  - Y\_Skew**: A text field containing "30".
  - Re Skew**: A button.
  - A vertical axis with a "+" sign at the top and a "V" sign at the bottom, with a double-headed arrow pointing up and down.

Figure 26. Property Table Form

Switch Board: "Agent Manager" button:

The "Agent Manager" button launches the "AgentList" form (Figure 27). This form allows the user to enter the parameters for the attackers (called agents in the form), specifically the start node, the target node and the target value.



Figure 27. Agent List Form

Switch Board: “Send to Xpress” button:

The IWD optimization model is currently implemented using Xpress-MP (therefore the name for this button, and associated form, as seen in Figure 28). The Xpress-MP form allows the user to enter the number of transparent interdictions, traps and decoys for the problem. In addition, an “index” is generated automatically as an identifier for the case (this field is required by the data tables associated with the optimization process to differentiate among several resource scenarios for the same network case).

The “*Obj. Value*” field is populated with the optimal (or otherwise achieved) objective function value after the IWD optimization model is run. That is, this field is the expected amount of target value interdicted assuming a perceptive behavior. Before running the IDW optimization model to retrieve the objective value (along with the full interdiction plan), the network and agents data needs to be sent to the optimization module. The “*Export Data*” button takes care of this function by translating the tables from the GUI’s database to the optimizer’s database. Once this step is complete, the “*Run Optimization*” button can be executed. This button will launch an MS-DOS batch file which runs the IWD optimization model in command line execution mode. When the execution is complete, the “*Update Results*” button can be used to read those results and transfer them to the GUI’s database. Finally, the “*Export to Simulation*” button transfers the data into text data files needed by the ANA simulation model.

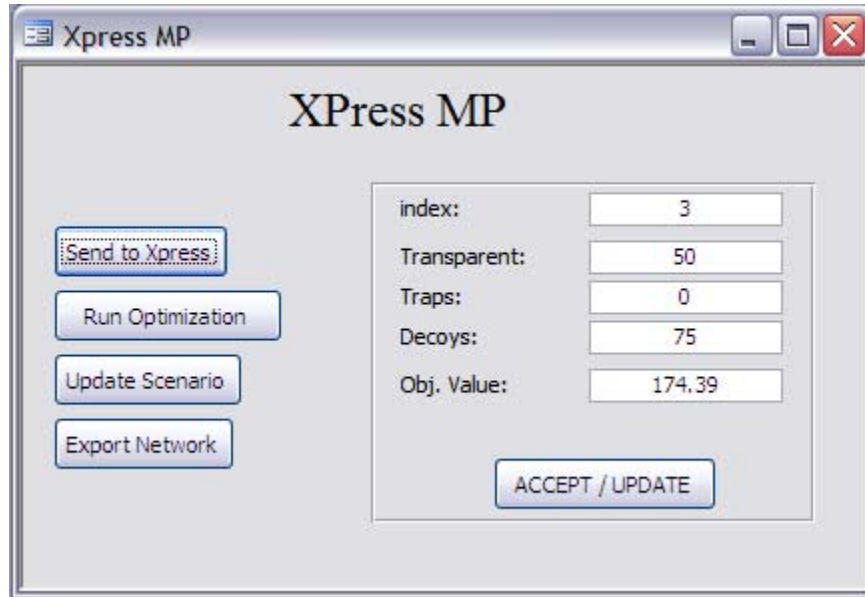


Figure 28. Xpress MP control form

Switch Board: “Simulation” button:

The ANA simulation can also be executed via the GUI. Figure 29 shows the “Simulation” form which takes care of the parameters and execution of the ANA simulation. The “Output File” field contains the file name where the simulation output will be redirected. The “*Number of Replications*” field is used to establish the number of samples for each simulation. The “*Communication Period*” parameter tells the ANA how long to extend the communication horizon during the communicative behavior simulation (please see Chapter II). The “Extra Parameters” field is an optional text field that allows the user to enter any other parameters (e.g., in anticipation of potential minor modifications to the simulation module that require a few additional input parameters).

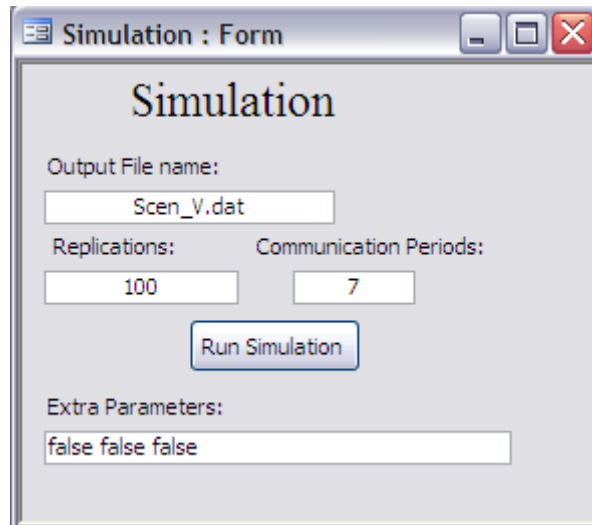
A screenshot of a Java window titled "Simulation : Form". The window has a title bar with standard minimize, maximize, and close buttons. The main content area is titled "Simulation" in a large, bold, serif font. Below the title, there are several input fields and a button. The first field is labeled "Output File name:" and contains the text "Scen\_V.dat". Below this are two fields: "Replications:" with the value "100" and "Communication Periods:" with the value "7". A blue button labeled "Run Simulation" is positioned below these two fields. At the bottom, there is a field labeled "Extra Parameters:" containing the text "false false false".

Figure 29. Simulation Form

Switch Board: “Network Drawer” button:

The “Network Drawer” is a Java applet [Savitch, 2005] which uses the node and arc text data files exported by the GUI as input for drawing the network.

If the networks to be drawn contain interdiction results in their data files, then the application will color the arcs by assigning blue to transparent interdictions, red to traps and yellow to decoys. All other arcs are colored in light gray. Also, agent starting positions nodes are colored red, and target nodes in black. Figure 30 shows an example of a network drawer display. The current version of the “Network Drawer” module does not depict agent routes.

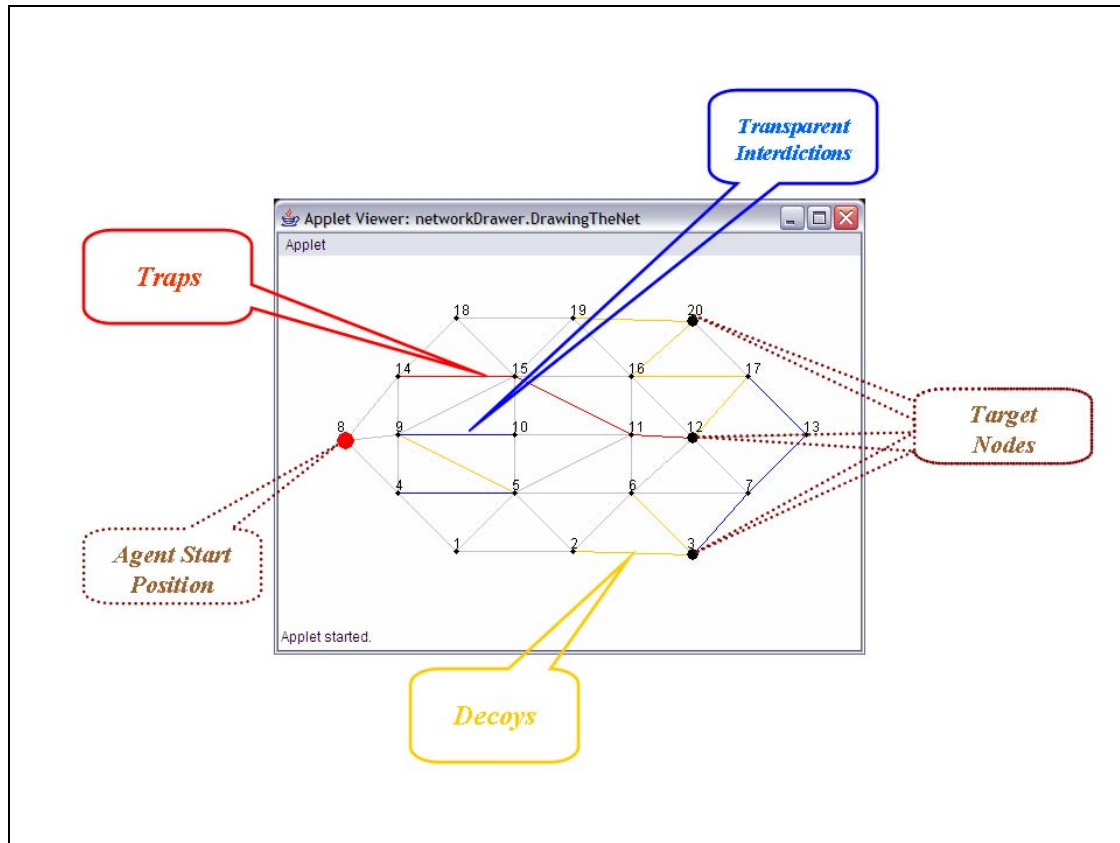


Figure 30. Network Drawer: Color Key

## B. SHAPE FILE TO ARC CONVERTER

Real-world networks are usually created using Geographic Information System (GIS) tools, using shape file format [ESRI, 1998] or similar. We have used a freeware converter tool called Shape Viewer 1.20 [Hammound, 1998], which allows us to convert “shapefile”-formatted files into Microsoft Excel and Access, so they can be utilized by our GUI with the help of some ad-hoc software code written in Java. As an illustrative example, in this section we discuss the necessary steps to convert the original Dystopia shape file [Locke, 2008]. (which originated the Dystopia case analyzed in Chapter III), into the format required by the GUI.

The first step consists of decoding the shape file into a set of line segment labels and coordinates. This is accomplished with Shape Viewer. Figure 31 shows the Dystopia road network file in Shape Viewer 1.20.

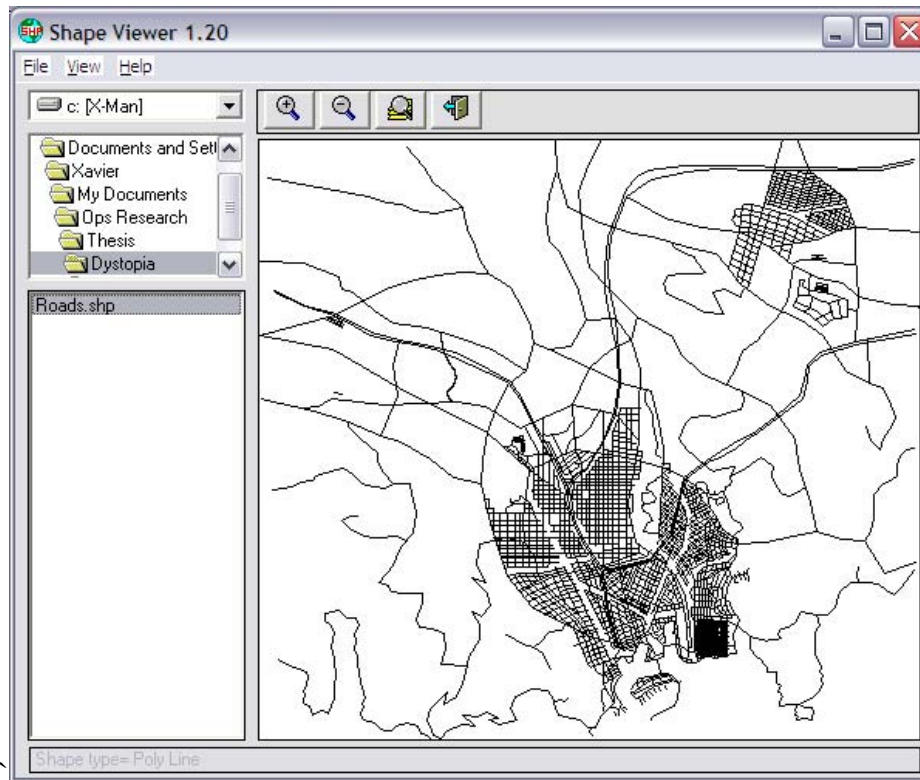


Figure 31. Visualization of Dystopia with Shape Viewer 1.20

A special feature of Shape Viewer 1.20 is its capability to export the shapefile file into an Excel document. The spreadsheet in Excel contains the basic structure necessary to reconstruct the shape file's picture. Table 13 shows a sample of the exported file in Excel. The columns labeled "polyline," "part" and "point id" uniquely identify the segment to which the point located at the associated X and Y coordinates belongs. By understanding this construct, a transformation of the table can be made, which converts these data into a "basic" set of arcs and nodes. The transformation is made in MS-Access. Table 14 shows an example of this new construct for the same data sample as in Table 13.

Polyline Id	Part Id	Point Id	X	Y
1	1	1	0.0755	0.1412
1	1	2	0.0689	0.1440
1	1	3	0.0590	0.1432
1	1	4	0.0542	0.1434
1	1	5	0.0483	0.1452
1	1	6	0.0417	0.1487
1	1	7	0.0368	0.1530
1	1	8	0.0414	0.1558
1	1	9	0.0457	0.1561
2	1	1	0.2334	0.3198
2	1	2	0.2363	0.3158

Table 13. Excerpt Output of Shape Viewer 1.20

NodeName	PartName	PolyID	Part	Point	X	Y	Tail	Head
1-1-1	1-1	1	1	1	7.55E-02	0.14123	1-1-1	1-1-2
1-1-2	1-1	1	1	2	6.89E-02	0.14396	1-1-2	1-1-3
1-1-3	1-1	1	1	3	5.90E-02	0.14321	1-1-3	1-1-4
1-1-4	1-1	1	1	4	5.42E-02	0.14336	1-1-4	1-1-5
1-1-5	1-1	1	1	5	4.83E-02	0.14515	1-1-5	1-1-6
1-1-6	1-1	1	1	6	0.04172	0.14871	1-1-6	1-1-7
1-1-7	1-1	1	1	7	3.68E-02	0.15302	1-1-7	1-1-8
1-1-8	1-1	1	1	8	4.14E-02	0.15584	1-1-8	1-1-9
1-1-9	1-1	1	1	9	4.57E-02	0.15614	1-1-9	NULL
2-1-1	2-1	2	1	1	0.2334	0.31975	2-1-1	2-1-2
2-1-2	2-1	2	1	2	0.23629	0.31578	2-1-2	2-1-3

Table 14. Basic Node and Arc Construct

An assumption at this point needs to be made regarding arc direction: We assume the allowed direction of flow is given by the order of the nodes defining the arc, from tail to head. The assumption allows us to arrange the complete list of nodes (3392 for Dystopia) in ascending order. Then, we can break the polylines by assigning a “NULL” value to the last point prior to the next polyline start node. This scheme allows us to build the set of nodes and arcs: The nodes simply consist of the “*Node Name*” and “*X,Y*” coordinates. The arcs consist of all (*Tail*, *Head*) pairs which do not contain “NULL” as an assignment to *Head*.

If, at this point, we redrew the map from the GUI, it would look just like the original shape file. However, the network would consist of no intersections, which are obviously necessary for network analysis. Figure 32 shows a section of Dystopia at this step of the transformation.

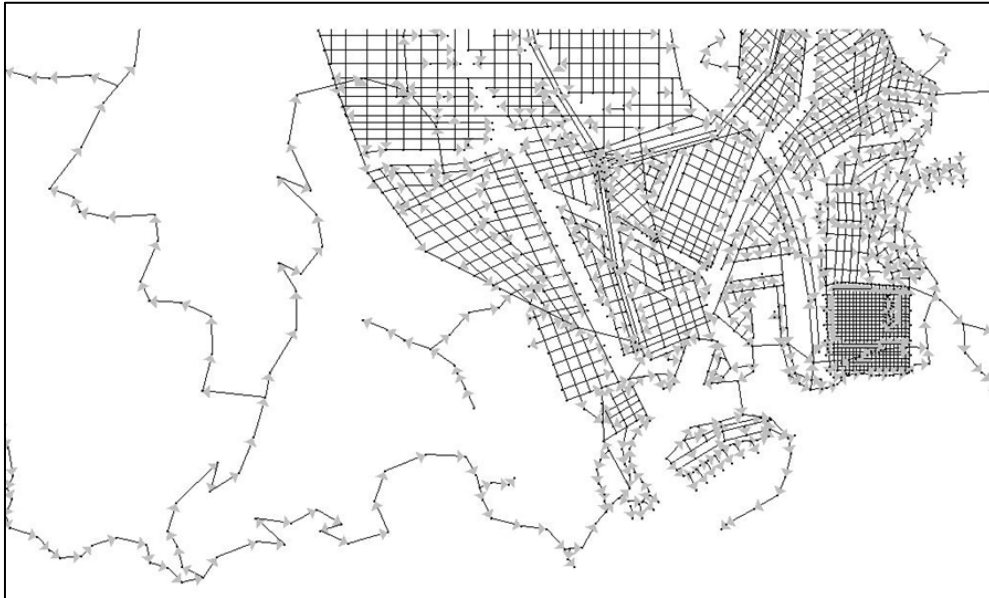


Figure 32. Dystopia (transformed) with no intersects

The calculation of the intersections requires specific code that finds the intersection point, creates the new nodes, determines the new arcs with proper directions, and inserts the arcs into the network. The code was written in Java and assumes that all the arcs that “intersect” are real intersections. In this sense, our Java code generates more intersections than exist in the actual network, for example, between highways and tertiary roads; and it does not take into account overpasses. These, however, represent a small fraction of all possible intersections, and therefore could be adjusted manually, if necessary. Figure 33 shows the same area portrayed in Figure 32, but with the intersections as found by the code marked in red.

The shape file converter may, in fact, have applications to any analysis which requires a translation from a shape file into a Node-Arc construct.

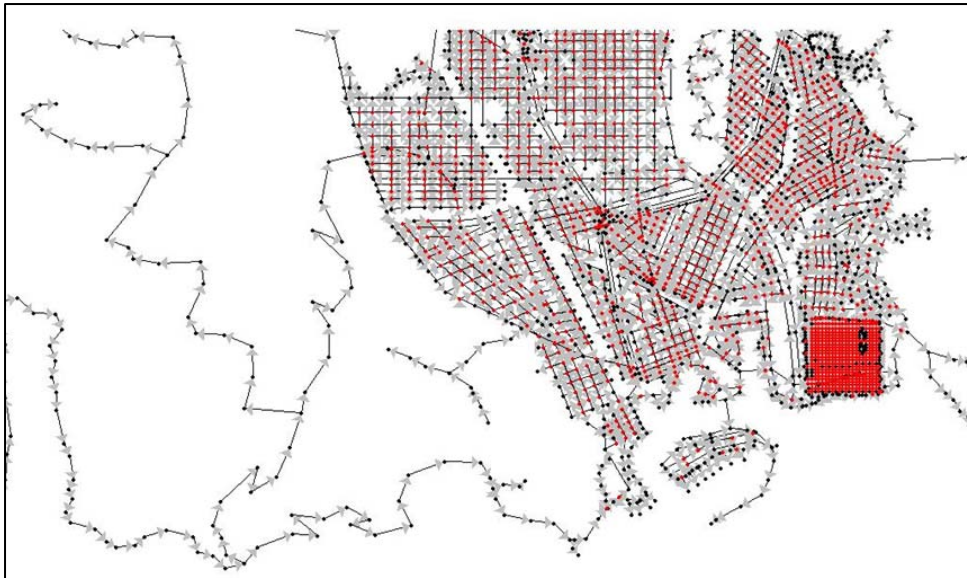


Figure 33. Final Transformation of Dystopia



## V. CONCLUSIONS AND FUTURE RESEARCH

### A. CONCLUSIONS

This thesis has initiated the testing of the IWD optimization model, which provides the distribution of transparent and deceptive interdiction assets in a network to defend against attacks carried out by VBIEDs. We measure the effectiveness of the allocation in terms of the probability of interdicting the attacker and/or the expected value interdicted (e.g., based on the expected number of lives saved).

In order to perform the testing of the IWD, the author has developed an agent-based simulation model, called ANA, and a graphical user interface which integrates the IWD and ANA modules.

The ANA module models multiple attacker behaviors which are run against the outputs of the IWD. Since the IWD model is based on the assumption of a *perceptive* behavior and optimal routing by the attackers, the ANA explores other five behaviors: *naïve*, *communicative*, *route blocker (static)*, *route blocker (dynamic)* and *clairvoyant*. These behaviors provide the defender with a more comprehensive set of measures to analyze the effectiveness of the IWD solution. The *naïve* behavior is solely based on shortest distance to the target. All other behaviors are still *perceptive*-like, but incorporate features such as learning (*communicative*), and random detours (*route blocker static* and *dynamic*). The *clairvoyant* behavior assumes full knowledge by the attacker and is used solely for bounding purposes.

Seven scenarios with different combinations of interdiction assets have been used in each of the small and realistically-sized network cases tested. Several outcomes of general application to any case have been derived, while other results are case-specific. We find most useful from a decision-making (attacker's or defender's) point of view the following insights:

- As classical network theory establishes, in a fully transparent scenario the *perceptive* behavior is the best for the attacker. Conversely, the only

reasonable defense against a *clairvoyant* attacker is to plan the allocation of all defenses as transparent.

- However, if the network incorporates deception, any other behavior (than *perceptive*) may be advantageous to the attacker. Which type of behavior is case-dependent. This can be utilized by either side to change their own tactics based on the information they have about the opponent's tactics. For example, if the attacker knows the defender is using deception (obviously, without identifying the specifics), a *naïve* route may be the best in some cases. Conversely, if the defender infers that the attacker is using a *naïve* behavior, he may decide to go back to a nearly transparent mode.
- A *communicative* behavior proves particularly effective over time for the attackers against scenarios containing traps. For the defender, this means it would be prudent to determine how much time to leave a deception plan in place before it loses efficiency. This thesis has illustrated how the tools developed can assist in determining this time frame.
- Decoys are most effective if used in defense against *perceptive*-like behaviors. They are rendered ineffective against attackers which behave *naïvely*. This is specifically noteworthy in scenarios where the only interdiction methods being utilized are deceptive (traps and decoys).
- Finally, if the defender expects the attacker to behave with any *perceptive*-like behavior, then the addition of transparent assets to traps and decoys may be of little value. This result, while case-dependent, could be significant to the defender since there is potential for cost savings via substitution of personnel at roadblocks by force-multiplier deceptive technologies.

## **B. FUTURE WORK**

While working on this thesis, several concepts have arisen that warrant further research. These relate to five areas:

(a) The IWD optimization model:

- The optimization model goal is to provide the defender with the best allocation of the interdiction assets that maximizes expected interdicted value (assuming a *perceptive* behavior). As stated in this thesis, the current IWD model accomplishes this only for the single-attacker case (with one or several targets). For a multiple-attacker case, the necessary simplifications for better tractability make the solution potentially suboptimal to the original problem. Future research may seek alternative methods to solve the original problem.

- The existing IWD model is a large-scale mixed-integer program, and becomes difficult to solve by commercial software in realistically-sized networks. Further research may develop new algorithms (probably based on decomposition) to overcome this difficulty.

(b) The ANA simulation model: Behaviors lend themselves to author's interpretation and assumptions. Developing more behaviors and creating mixed-ones (e.g., a *naïve-communicative*, or “weighed” behaviors) can add realism and cultural customization to the attackers.

(c) The interaction between IWD and ANA models: Since the IWD model assumes a *perceptive* behavior, feedback from the ANA simulation might guide subsequent optimizations. Feedback may consist, for example, on the simulated routes for the attackers, which can be incorporated into a modified version of IWD to produce alternative interdiction plans. For example, the modified IWD could assign more “weight” to arcs identified by the simulation.

(d) Testing:

- Further testing is necessary, including sensitivity analysis on the amount of assets of all types. This may, for example, help answer the question about minimal combinations of assets necessary to obtain a pre-specified interdiction value required for mission success.

- Testing on a real set of city networks, and using accurate values for the interdiction probabilities (nominal and for each interdiction asset type) would add realism to the results. However, even in the absence of a real data set, sensitivity analysis over those values may still provide useful insights. For example, it might be determined that a certain allocation of traps and decoys remains optimal if the interdiction probabilities of traps and the perceptive probabilities of decoys lie within certain ranges.

- A robust design of experiments analysis can be developed with the goal of finding defensive strategies that hold up against a variety of attack behaviors. “Robustness” here entails measuring deviations from each behavior outcome with respect to an ideal (or desired) interdiction value.

(e) Graphical user interface: Improving the GUI for ease of use will mature the tool to a state where it can be delivered to on-scene commanders. For example, a mouse driven editor and selector in the Network Drawer application can help with network generation and manipulation.

## APPENDIX:     DYSTOPIA GRAPHICS

This Appendix provides a graphical representation of the IWD solution for the Dystopia test case. Red color indicates placement of traps, yellow is used for decoys and blue represents transparent interdictions.

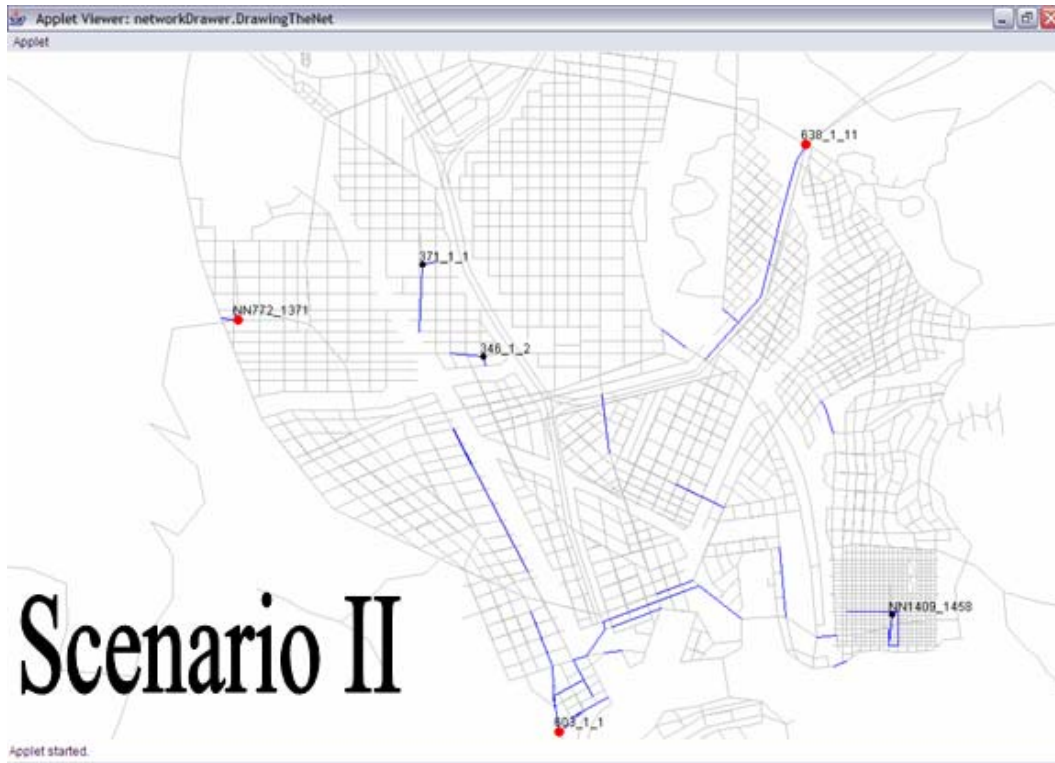


Figure 34.

Figure 35.     Dystopia Scenario II, Transparent Interdiction Only

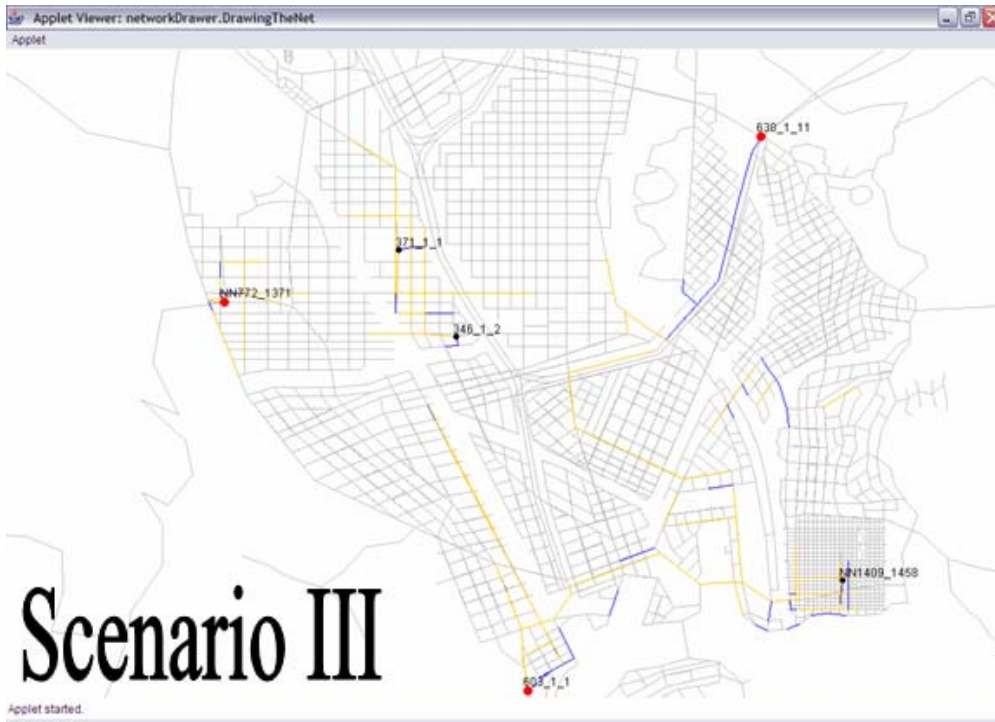


Figure 36. Dystopia Scenario III, Transparent Interdiction and Decoys

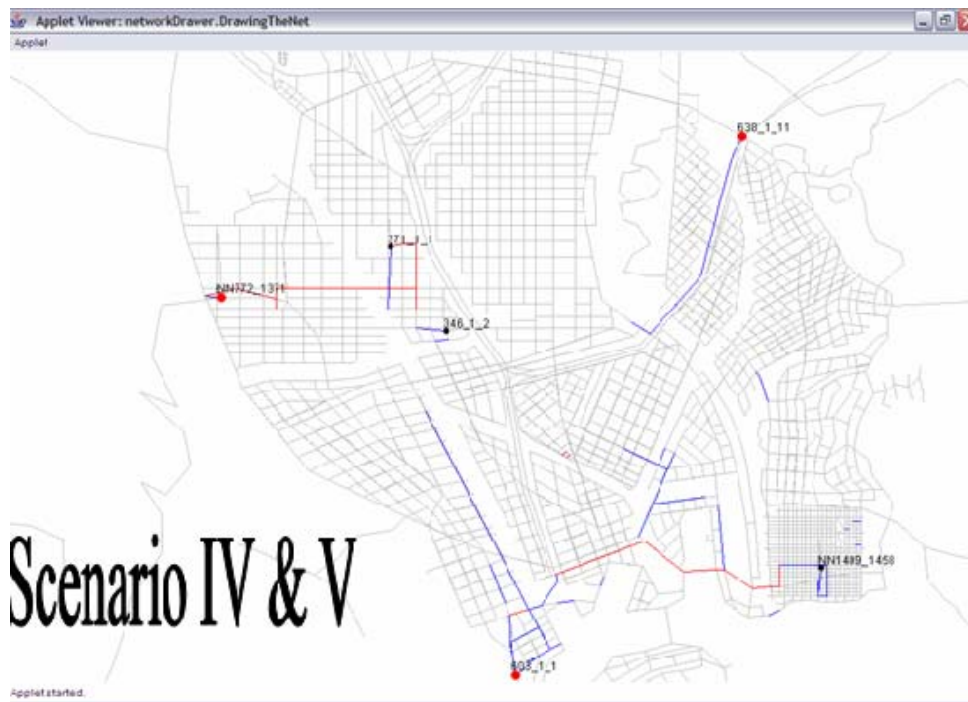


Figure 37. Dystopia Scenario IV & V, Transparent Interdictions and Traps

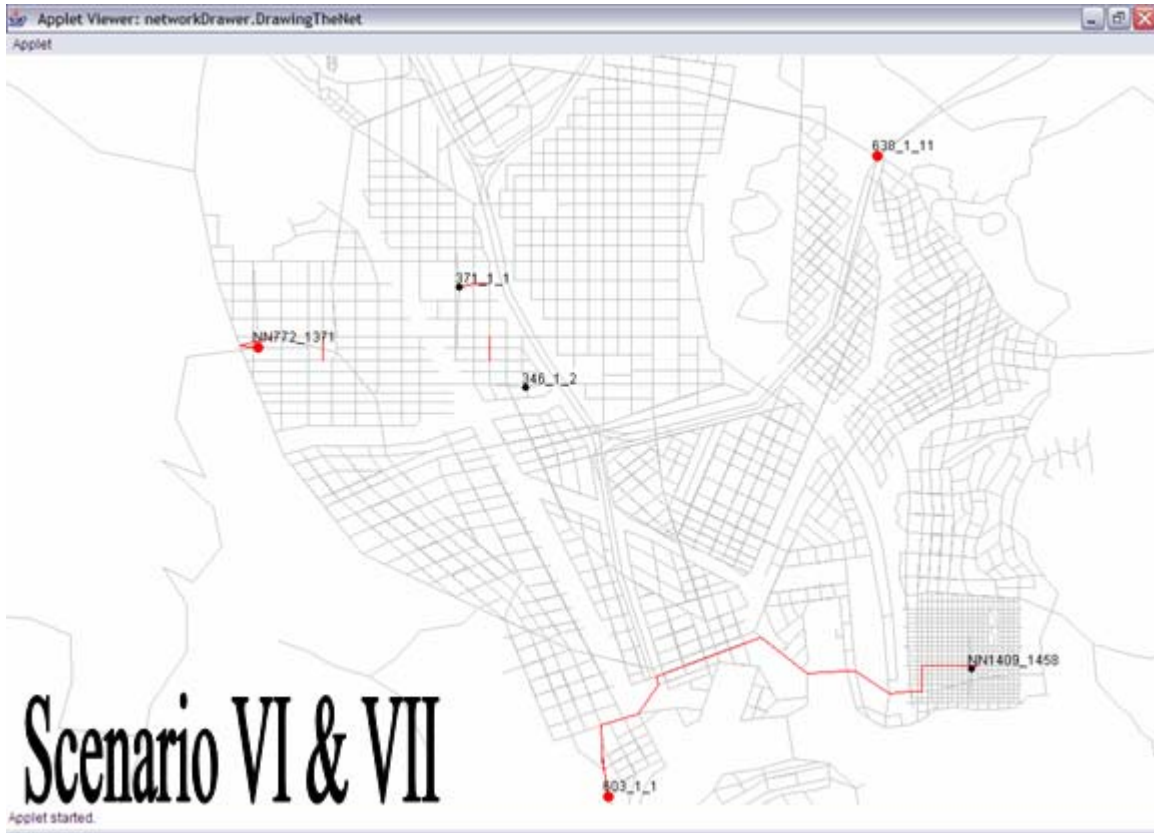


Figure 38. Dystopia Scenario VI & VII, Traps only

THIS PAGE INTENTIONALLY LEFT BLANK



## LIST OF REFERENCES

- Ahuja, R.K, Magnanti, T.L, and Orlin, J.B. (1993). *Network Flows*, Prentice Hall, Upper Saddle River, NJ.
- Brown, G., Carlyle, M., Salmeron, J. and Wood, K. (2006). "Defending Critical Infrastructure," *Interfaces*, Vol. 36, 530-544.
- BATF, U.S. Bureau of Alcohol Tobacco and Firearms (2005). BATF Explosive Standards. (Accessible via Wikipedia at <http://en.wikipedia.org/wiki/Image:Vbied-standards-chart.jpg>. (accessed March 2008).
- Brown, G., Carlyle, W.M., Salmeron, J. and Wood, K. (2005) "Analyzing the Vulnerability of Critical Infrastructure to Attack, and Planning Defenses," in *Tutorials in Operations Research: Emerging Theory, Methods, and Applications*, H. Greenberg and J. Smith, eds., Institute for Operations Research and Management Science, Hanover, MD.
- Dash Optimization (2007). [www.dashoptimization.com](http://www.dashoptimization.com) (accessed March 2008).
- Dudonis, K. J. (2005). *The Counterterrorism Handbook*, CRC Press.
- ESRI, Environmental Systems Research Institute (1998). "ESRI Shapefile Technical Description," white paper.
- Hammoud, M. (1998). Shapeviewer 1.20. [www.shapeviewer.com](http://www.shapeviewer.com) (accessed March 2008).
- Locke, J. (2008). Dystopia: A fictional city for homeland security simulation (unpublished map). MOVES Institute, Naval Postgraduate School, Monterey, CA.
- Main, M. (2006). *Data Structures and Other Objects using Java*, Pearson Addison Wesley.
- Microsoft® Corporation (2003). <http://www.microsoft.com> (accessed March 2008).
- Motto, A.L., Arroyo, J.M. and Galiana, F.D. (2005). "A Mixed-Integer LP Procedure for the Analysis of Electric Grid Security under Terrorist Threat," *IEEE Trans. on Power Systems*, vol. 20, No. 3, pp. 1357-1365.
- Salmeron, J. (2007). "Interdiction with Deception," working report, Naval Postgraduate School.
- Savitch, W. J. (2005). *Java: An Introduction to Problem Solving and Programming*, Pearson Prentice Hall, Upper Saddle River, NJ.

Spiller, R. J. (1992). "Combined Arms in Battle Since 1939," U.S. Army Command and General Staff College Press.

Wood, R.K. (1993). "Deterministic Network Interdiction," *Mathematical and Computer Modelling*, 17, 1-18.

## **INITIAL DISTRIBUTION LIST**

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. COL Andy Hernandez  
Naval Postgraduate School  
Monterey, California
4. COL Saverio Manago  
Center for Army Analysis  
Ft. Belvoir, Virginia